PENGELOMPOKAN FUNGSI AKTIF SENYAWA DATA SMILES (*Simplified Molecular Input Line Entry System*) MENGGUNAKAN METODE K-MEANS DENGAN INISIALISASI PUSAT KLASTER MENGGUNAKAN METODE *HEURISTIC O(N LOGN)*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Sherly Witanto
NIM: 145150201111160

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PENGELOMPOKAN FUNGSI AKTIF SENYAWA DATA SMILES (*Simplified Molecular Input Line Entry System)* MENGGUNAKAN METODE K-MEANS DENGAN INISIALISASI PUSAT KLASTER MENGGUNAKAN METODE HEURISTIC O(N LOGN)

SKRIPSI

Untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun oleh:
Sherly Witanto
NIM: 145150201111160

Skripsi ini telah diuji dan dinyatakan lulus pada:

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Dian Eka Ratnawati , S.Si, M.Kom	Syaiful Anam, S.Si.,MT.,Ph.D
NIP. 19730619 200212 2 001	NIP. 19780115 200212 1 003

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, didalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70)

Malang, 20 Juni 2018

Sherly Witanto

NIM: 145150201111160

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang selalu melimpahkan rahmat dan karunia-Nya kepada penulis sehingga penulis dapat menyelesaikan skripsi dengan judul "PENGELOMPOKAN FUNGSI AKTIF SENYAWA DATA SMILES (*Simplified Molecular Input Line Entry System*) MENGGUNAKAN METODE K-MEANS DENGAN INISIALISASI PUSAT KLASTER MENGGUNAKAN METODE HEURISTIC O(N LOGN)" sebagai syarat dalam memperoleh gelar sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya. Dalam proses penyusunan skripsi penulis mendapatkan bantuan berupa moral maupun materil dari banyak pihak. Maka dari itu, penulis ingin mengucapkan rasa terima kasih sebanyak-banyaknya kepada:

1. Ibu Dian Eka Ratnawati , S.Si, M.Kom selaku dosen pembimbing I yang telah memberikan arahan, masukan ilmu dan bimbingan sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak Syaiful Anam, S.Si.,MT.,Ph.D selaku dosen pembimbing II yang juga turut memberikan arahan, masukan ilmu dan bimbingan sehingga penulis dapat menyelesaikan skripsi ini.
3. Bapak dan Ibu dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan ilmu selama penulis melaksanakan kegiatan perkuliahan di kampus tercinta ini.
4. Staff dan karyawan Fakultas Ilmu Komputer yang telah membantu proses selama perkuliahan dan penulisan skripsi ini.
5. Papa Jongki Witanto, S.H. yang telah memberikan dukungan dalam segala aspek sehingga saya dapat menyelesaikan kuliah saya.
6. Mama Swan yang telah membesarkan dan merawat saya sehingga dapat menyelesaikan tugas belajar saya dan menjadi anak yang lebih baik.
7. Sonny Witanto, Sheren Witanto dan Emak yang telah memberikan semangat dan dukungan dalam menyelesaikan tugas skripsi ini.
8. Koko Jerry yang telah segenap hati membantu dalam segala situasi, memberikan dukungan dan doa sehingga saya dapat menyelesaikan skripsi ini tepat waktu.
9. Teman-teman seperjuangan dalam menyusun skripsi Suhhy Ramzini, Muhammad Iskandar A.R, Nur Khilmiyatul, R. Rizky Widdie, Nyimas Ayu Widi Indriana, Yunita Dwi Alfianti yang telah saling membantu dan berdiskusi dalam penyelesaian skripsi ini.
10. Teman-teman Informatika FILKOM UB 2014 yang telah memberikan kesan dan pesan selama penulis menempuh perkuliahan di Fakultas Ilmu Komputer Universitas Brawijaya.
11. Teman Jonggol Dhika Rozqi Anggitama, Kevin Dwiki Saputra, Hermawan Wijaya, David Christanto, Yosua Tito Sumbogo, R. Rizky Widdie, Maria Rantikasari, Reka Suryani Sidauruk yang telah memberikan pengalaman, diskusi dan motivasi selama penullis melaksanakan perkuliahan di Fakultas Ilmu Komputer Universitas Brawijaya.

12. Serta semua phiak yang tidak bisa penulis sebutkan satu per satu yang juga turut memberikan dukungan, doa, semangat dan motivasi kepada penulis.

Penulis menyadari betul bahwa skripsi ini masih memiliki kekurangan dan jauh dari kata sempurna. Oleh sebab itu, penulis mengharapkan adanya saran dan kritik guna membangun dan memperbaiki kekurangan yang penulis punya. Akhir kata, penulis sangat berharap skripsi ini bisa memberikan manfaat kepada semua pihak.

Malang, 20 Juni 2018

Sherly Witanto

sherlywitanto@gmail.com

ABSTRAK

Sherly Witanto. 2018. Pengelompokan Fungsi Aktif Senyawa Data SMILES (*Simplified Molecular Input Line Entry System)* menggunakan Metode K-Means dengan Inisialisasi Pusat Klaster menggunakan Metode *Heuristic O(N LogN).* Skripsi Program Studi Teknik Informatika / Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya

Pembimbing: Dian Eka Ratnawati , S.Si, M.Kom dan Syaiful Anam, S.Si.,MT.,Ph.D

Senyawa aktif mempunyai salah satu kegunaan sebagai bahan obat-obatan yang mampu mencegah maupun menyembuhkan penyakit. Sebagian senyawa aktif sudah ditemukan fungsinya dan sebagian lagi masih dalam tahap penelitian. Saat ini di Indonesia masih belum ada program yang mampu mengklasifikasi senyawa kimia sebagai obat untuk penyakit tertentu. Notasi SMILES merupakan konversi senyawa kimia dalam bentuk notasi baris. Notasi SMILES mampu memberikan kemudahan pada proses komputerisasi pada klasifikasi senyawa kimia. Klasifikasi atau pengelompokan notasi SMILES dilakukan dengan mengambil nilai 11 fitur atom B,S,N,O,I,F,C,P,Cl,Br dan OH yang ada pada senyawa tersebut. Sebelum diproses, untuk mendapatkan nilai fitur dilakukan proses dengan membagi masing-masing jumlah atom dengan panjang senyawanya.

Algoritme K-Means merupakan metode klastering yang paling banyak digunakan karena bersifat mudah dan sederhana. Pengelompokan fungsi aktif menggunakan metode K-Means mempunyai kelemahan pada proses inisialisasi klaster yang bersifat *random,* sehingga digunakan metode *heuristic o(n logn)* untuk mendapatkan inisial klaster dengan nilai yang lebih baik. Berdasarkan perangkat lunak yang telah dibuat, pengujian dilakukan dengan menggunakan data latih sebanyak 512 dan data uji sebanyak 128. Akurasi yang diperoleh dari pengujian yaitu sebesar 63% dan pengujian menggunakan *K-Fold Cross Validation* dengan 10 kali pengujian menghasilkan akurasi rata-rata sebesar 52.58%. Pengujian menggunakan K-Means dengan *heuristic o(n logn)* menghasilkan akurasi yang lebih baik dibandingkan dengan K-Means konvensional.

Kata Kunci: SMILES, K-Means, *Heuristic O(N LogN)*

ABSTRACT

Active compounds have function as a medicine that can prevent or cure diseases. Some of the active compounds have been known the function and some are still in the research stage. Currently in Indonesia there is still no program that capable to classifying chemical compounds as drugs for certain diseases. SMILES notation is the conversion of chemical compounds in the form of line notation. Notation SMILES able to provide convenience to the process of computerization on the classification of chemical compounds. The classification of the SMILES notation is carried out by taking the values of the B, S, N, O, I, F, C, P, Cl, Br and OH atoms present in the compound. Before being processed, to get the value of the feature is done by dividing the process of each atom with the length of the compound.

K-Means algorithm is the most widely used clustering method because it is easy and simple. The grouping of active function using K-Means method has weakness in random cluster initialization process, so that heuristic method o (n logn) is used to get the cluster initials with better value. Based on the software that has been made, the test is done using 512 of training data and test data as much as 128. Accuracy obtained from the test that is equal to 63% and testing using ¬K-Fold Cross Validation with 10 times the test produces an average accuracy of 52.58 %. Testing using K-Means with heuristic o (n logn) yielded better accuracy compared to conventional K-Means.

***Keywords**: SMILES, K-Means, Heuristic O(N LogN)*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

SOURCE CODE

BAB 1 PENDAHULUAN

1.1 Latar belakang

Dalam bidang kimia, seringkali kita mendengar istilah atom, molekul dan ion. Pada umumnya orang mendengar istilah tersebut tanpa tahu artinya. Istilah tersebut dapat saling berkaitan dan membentuk senyawa. Senyawa adalah zat tunggal yang terdiri dari dua atau lebih unsur yang berbeda dan membentuk ikatan, sehingga terbentuklah senyawa sebagai zat baru yang mempunyai fungsi tertentu. Senyawa dibagi menjadi dua, yaitu senyawa aktif dan tidak aktif. Senyawa aktif mempunyai farmakologis yang berfungsi sebagai obat tertentu. Sedangkan senyawa tidak aktif tidak mempunyai peran signifikan dan hanya berfungsi sebagai zat tambahan/pengikat. Sebagian senyawa aktif sudah ditemukan fungsinya dan sebagian lagi belum ditemukan dan masih dalam tahap penelitian. (Rizki *et al,* 2015).

Bagi orang kimia, untuk mengetahui kegunaan suatu senyawa diperlukan suatu penelitian yang memerlukan biaya dan waktu yang tidak sedikit, salah satu contohnya adalah dengan metode ekstraksi (Mukhriani, 2016). Sehingga diperlukan senyawa dengan bentuk yang mudah untuk dipahami yaitu dengan menggunakan kode SMILES (*Simplified Molecular Input Line Entry System*) untuk membantu pengelompokan senyawa kimia. Kode SMILES mengkonversi senyawa dalam bentuk notasi baris untuk menggambarkan senyawa kimia. Kode SMILES mempunyai kelebihan yaitu efisien dalam penyimpanan data dan komputasi, sehingga dapat digunakan pengelompokan oleh bidang IT (Weininger, 1987).

Senyawa aktif merupakan bahan obat yang berguna bagi pencegahan maupun penyembuhan penyakit tertentu dan terdapat pada tumbuhan maupun hewan (Salni *et al*, 2011). Saat ini di Indonesia masih belum ada program yang dapat mengklasifikasi senyawa kimia sebagai obat untuk penyakit tertentu. Masih sedikit orang yang mengetahui notasi SMILES dan bagaimana melakukan pengelompokan senyawa tersebut. Notasi SMILES sendiri dapat memberikan kemudahan pada proses komputerisasi dan penyimpanan data di komputer supaya mempermudah pengelompokan senyawa kimia. Dari seluruh senyawa kimia yang ada, sebagian sudah ditemukan kegunaannya, namun sebagian lagi masih dalam tahap pengujian. Jumlah senyawa kimia yang tidak sedikit memerlukan proses pengelompokan senyawa yang dapat mempermudah pencarian fungsi yang terkandung di dalamnya, yaitu sebagai obat suatu penyakit tertentu.

K-Means merupakan metode klastering paling umum, sederhana dan mudah. Hal ini dikarenakan K-Means dapat mengelompokkan data dalam jumlah banyak dan dalam waktu yang relatif cepat. Berbeda dengan K-Means konvensional yang inisialisasi pusat klasternya dilakukan secara *random*, K-Means dengan inisialisasi klaster menggunakan metode *heuristic o (n logn)* atau disebut juga *improved* K-Means melakukan inisialisasi pusat klaster dengan distribusi data (Nazeer, 2011). Pusat klaster digunakan untuk mencari jarak terpendek antar data dengan pusat klaster itu sendiri, sehingga dapat menjadi anggota klaster tersebut.

Algoritme K-Means konvensional memiliki kelemahan dikarenakan pembentukan awal pusat klaster yang bersifat *random.* Hasil dan kecepatan proses pengelompokan data bergantung pada inisialisasi tersebut, sehingga untuk menghindari hasil yang kurang maksimal diperlukan metode yang membantu dalam penentuan inisial awal pusat klaster (Tahta, 2012).

Pada penelitian sebelumnya oleh K A Nazeer Abdul, S D Madhu Kumar dan M P Sebastian dengan judul *Enchancing the K-Means Clustering Algorithm by using a O(n logn) Heuristic Method for Finding Better Initial Centroids* menggunakan *dataset e-coli, breast cancer* dan *thyroid.* Penelitian tersebut menghasilkan akurasi yang lebih baik dan pemrosesan lebih cepat dibandingkan dengan K-Means konvensional (Nazeer, 2011).

Penelitian yang dilakukan oleh Rinadewi Astuti berjudul Implementasi Algoritme *K-Means Clustering* dengan Inisialisasi *Centroid* Menggunakan Metode *Heuristic O(N logn)* menggunakan *dataset iris*. Penellitian tersebut menghasilkan akurasi yang lebih baik dan waktu pemrosesan yang lebih cepat dibandingkan K-Means konvensional (Astuti, 2015).

Algoritme K-Means klastering akan digunakan untuk mengklasterisasi data notasi SMILES sehingga diharapkan dapat mengenali kegunaan senyawa aktif dari fitur yang didapat dari ciri senyawa tersebut. Inisialisasi pusat klaster menggunakan metode *heuristic n (o logn)* untuk dibandingkan dengan metode k means konvensional. Fitur yang dapat digunakan meliputi panjang kode SMILES dan jumlah tiap elemen. Fitur tersebut diharapkan mampu memproses algoritme K-Means pada sejumlah klaster. Metode ini diharapkan mampu mengidentifikasi kegunaan senyawa pada kode SMILES.

Berdasarkan alasan yang diperoleh, maka dibuat solusi yaitu program yang akan mengelompokkan fungsi senyawa aktif yang berguna untuk mengetahui kegunaan obat dari senyawa tersebut. Program ini dibuat dengan menerapkan metode klasterisasi K-Means dengan inisialisasi pusat klaster menggunakan metode *heuristic o (n logn)* dengan objek notasi senyawa aktif SMILES.

1.2 Rumusan masalah

Rumusan masalah yang didapat dalam penelitian yaitu:

1. Bagaimana melakukan inisialisasi pusat klaster pada metode K-Means menggunakan metode *heuristic o (n logn)*?
2. Bagaimana hasil akurasi pengelompokan fungsi senyawa aktif kode SMILES menggunakan algoritme K-Means yang inisial pusat klasternya telah ditentukan dengan metode *heuristic o (n logn)* dengan algoritme K-Means konvensional?

1.3 Tujuan

Adapun tujuan penelitian yang didapat dalam penelitian ini yaitu:

1. Menerapkan metode K-Means dengan inisial pusat klaster menggunakan metode *heuristic o (n logn)* untuk mengelompokkan fungsi senyawa aktif kode SMILES.
2. Mengetahui hasil akurasi pengelompokan fungsi senyawa aktif kode SMILES menggunakan algoritme K-Means yang inisial pusat klasternya telah ditentukan dengan metode *heuristic o (n logn)* dengan algoritme K-Means konvensional.

1.4 Manfaat

Diharapkan penelitian ini dapat menghasilkan sistem yang mampu menerapkan metode K-Means dan *heuristic o (n logn)* dan mampu memperoleh perbandingan hasil terbaik antara algoritme K-Means konvensional dengan algoritme K-Means yang inisialisasi pusat klasternya menggunakan metode *heuristic o (n logn).* Penelitian ini dapat berguna bagi peneliti khususnya bidang kimia dan Teknik Informatika untuk penelitian selanjutnya.

1.5 Batasan masalah

Adapun batasan masalah yang digunakan dalam penelitian agar penelitian dapat lebih terfokus, yaitu:

1. Implementasi K-Means berfokus pada inisial awal pusat klaster.
2. Senyawa yang digunakan merupakan senyawa aktif yang mempunyai fungsi untuk penyembuhan penyakit tertentu dan diambil dari https://pubchem.ncbi.nlm.nih.gov .
3. Fitur *input* yang digunakan adalah jumlah masing-masing elemen penyusun dan panjang kode SMILES.
4. Jumlah kelas yang digunakan adalah dua.
5. Hasil yang dicari berupa pengelompokan fungsi senyawa aktif serta perbandingan tingkat akurasi.
6. Sistem dibuat berbasis web dengan bahasa pemrograman PHP dan basis data MySQL.

1.6 Sistematika Pembahasan

Sistematika penulisan laporan penelitian ini dijelaskan sebagai berikut.

BAB I. PENDAHULUAN

Pada bab pendahuluan, penulis membahas tentang alasan dilakukannya penelitian dan penulisan.

BAB II. LANDASAN KEPUSTAKAAN

Pada bab landasan pustaka, penulis menjelaskan konsep-konsep yang digunakan untuk merancang sistem pengelompokan fungsi penyakit dengan

metode K-Means yang inisial pusat klasternya menggunakan metode *heuristic o (n logn)*.

BAB III. METODOLOGI

Pada bab metodologi, penulis menjelaskan metodologi penelitian yang digunakan untuk merancang sistem klasifikasi fungsi penyakit dengan metode K-Means yang inisial pusat klasternya menggunakan metode *heuristic o (n logn)*.

BAB IV. PERANCANGAN

Pada bab perancangan, berisi perancangan sistem, perhitungan manual menggunakan metode K-Means dengan inisial klaster menggunakan metode *heuristic o (n logn)* serta perancangan antarmuka. Penulis menjelaskan analisis kebutuhan perangkat yang akan digunakan, yaitu perangkat keras dan lunak. Bab ini juga berisi antarmuka dan implementasi metode K-Means dengan inisial pusat klaster *Heuristic O(N LogN)* pada sistem.

BAB V. HASIL

Pada babwhasil, penulis menuliskan spesifikasi yang digunakan, serta hasil implementasi sistem yang telah dibuat.

BAB VI. PEMBAHASAN

Pada bab pembahasan berisi hasil pengujian, akurasi perbandingan dan analisis dari sistem yang telah dibuat.

BAB VII. PENUTUP

Pada bab ini berisi kesimpulan yang diperoleh dari pembuatan, pengujian dan analisis serta saran dari penulis untuk penelitian lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Kajian pustaka merupakan referensi yang digunakan sebagai pendukung metode yang digunakan yaitu K-Means dengan inisial klaster menggunakan metode *heuristic o(n logn)*. Referensi didapatkan dari penelitian sebelumnya yang menggunakan metode serupa dengan penelitian yang dilakukan. Analisis kajian pustaka akan dijelaskan pada Tabel 2.1.

Tabel 2.1 Kajian Pustaka

No	Judul	Obyek (Masukan)	Metode (Proses)	Hasil (Keluaran)
1	Enhancing the K-means Clustering Algorithm by Using a O(n logn) Heuristic Method for Finding Better Initial Centroids	Data *e-coli, breasts cancer,* dan *thyroid.*	Membandingkan akurasi dan waktu dari metode K-Means modifikasi dengan K-Means konvensional	Hasil dari K-Means modifikasi lebih baik dengan hasil akurasi data *e-coli* 81,5% dengan waktu 40 mili detik, pada data *breast cancer* 96,2% dengan waktu 42 mili detik sedangkan data *thyroid* 86% dengan waktu 52 mili detik
2	Implementasi Algoritme *K-Means Clustering* dengan Inisialisasi *Centroid* Menggunakan Metode *Heuristic O(N logn)*	*Dataset Iris*	Membandingkan akurasi dan waktu pemrosesan metode K-Means konvensional dan *Improved* K-Means	Hasil penelitian menunjukkan bahwa akurasi dan waktu pemrosesan *improved* K-Means mempunyai hasil yang lebih baik dibandingkan dengan K-Means konvensional

Penelitian yang berjudul "Enhancing the K-means Clustering Algorithm by Using a O(n logn) Heuristic Method for Finding Better Initial Centroids" dilakukan oleh K A Nazeer Abdul, S D Madhu Kumar dan M P Sebastian pada tahun 2011. Penelitian dilakukan dengan tujuan mengetahui hasil perbandingan metode K-Means modifikasi dengan K-Means konvensional. K-Means modifikasi mencapai akurasi rata-rata lebih baik yaitu sebesar 87,9% dibandingan dengan K-Means konvensional dengan rata-rata akurasi sebesar 83,57%. Sedangkan waktu eksekusi pada K-Means modifikasi mencapai rata-rata 44,66 mili detik dibandingkan

dengan K-Means konvensional yang memerlukan waktu lebih lama yaitu rata-rata 64 mili detik (Nazeer, 2011).

Penelitian yang berjudul "Implementasi Algoritme *K-Means Clustering* dengan Inisialisasi *Centroid* Menggunakan Metode *Heuristic O(N logn)*" dilakukan oleh Rinadewi Astuti pada tahun 2015. Penelitian mengacu pada penelitian sebelumnya yang dilakukan oleh Nazeer yaitu membandingkan metode K-Means konvensional dan K-Means modifikasi. Penelitian dilakukan menggunakan *dataset iris*. Hasil yang didapat dari penelitian menunjukkan bahwa *improved* K-Means mempunyai akurasi dan waktu proses yang lebih baik dibandingkan dengan K-Means konvensional (Astuti, 2015).

2.2 Pre-processing

Preprocessing pada notasi SMILES dilakukan menggunakan *regular expression* (*regex*). *Preprocessing* bertujuan membentuk struktur data sesuai kebutuhan, serta untuk mengetahui letak huruf atau kata (*the number of terms*) (Manning et al, 2009). Sedangkan *regex* merupakan sebuah rumusan yang berguna untuk melakukan pencarian terhadap pola suatu kalimat (Muliantara, 2009). Pola umum penggunaan *regex* akan dijelaskan pada Tabel 2.2.

Tabel 2.2 Pola Umum Penggunaan *Regular Expression* (*regex*)

Pola	Penjelasan
[]	Kurung siku merupakan pola yang cocok dengan satu karakter yang ada dalam kurung. Misalkan terdapat pola "a[bcd]i", maka akan cocok dengan kata "abi", "aci", dan "adi". Apabila ingin mencocokkan pada semua huruf, maka bisa dibuat pola "[a-z]" yang artinya akan cocok dengan semua huruf kecil dari "a" sampai "z". Apapun yang dimasukkan ke dalam kurung siku, maka akan menjadi objek yang dicari atau dicocokkan.
[^]	Pola ini cocok dengan karakter yang berada dalam kurung siku. Misalkan terdapat pola "*[^abc]*", maka artinya adalah cocok dengan semua karakter kecuali "a", "b", dan "c".
?	Tanda tanya akan cocok dengan nol atau satu karakter sebelumnya. Misalkan terdapat pola "*kamu?*", maka akan cocok dengan kata "kam" atau "kamu".
+	Tanda tambah cocok dengan satu atau lebih karakter yang sebelumnya. Misalkan terdapat pola "*ayo+k*", maka akan cocok dengan kata "ayok", "ayook", "ayoook", dan seterusnya (dengan jumlah karakter "o" tidak terbatas).
*	Tanda bintang akan cocok dengan nol atau lebih karakter sebelumnya. Misalkan terdapat pola "*ku*y*", maka akan cocok dengan kata "ky", "kuy", "kuuy", dan seterusnya (dengan jumlah karakter "u" tidak terbatas).

Tabel 2.2 (Lanjutan)

Pola	Penjelasan
{x}	Pola yang akan cocok dengan karakter sebelumnya sebanyak "x". Misalkan terdapat pola "*[0-9]{3}*", maka akan cocok dengan semua bilangan yang berjumlah 3 digit.
(x,y)	Pola yang akan cocok dengan karakter sebelumnya sebanyak "x". Misalkan terdapat pola "*[0-9]{3, 5}*", maka akan cocok dengan semua bilangan yang berjumlah di antara 3 sampai 5 digit.
!	Apabila diletakkan pada bagian depan pola, maka memiliki arti "bukan". Misalkan terdapat pola "*!aki*", maka akan cocok untuk semua kata namun buka kata "aki" (dengan jumlah karakter sebanyak 3).
^	Cocok untuk kata atau karakter yang berada di awal. Misalkan terdapat pola "*^aku*" dan terdapat kalimat "aku adalah aku yang bukan raja", maka akan cocok pada kata "aku" yang berada di awal kalimat saja.
$	Cocok untuk kata atau karakter yang berada di akhir. Misalkan terdapat pola "*$raja*" dan terdapat kalimat "aku adalah raja dari semua raja", maka akan cocok pada kata "raja" yang berada di akhir kalimat saja.
()	Tanda kurung digunakan untuk membuat grup yang mana akan terjadi pengelompokkan karakter-karakter menjadi *single unit* (satu kesatuan).
\	Garis miring digunakan untuk mengawali penggunaan *regex*.

Preprocessing notasi SMILES dengan *regex* adalah dengan mencari dan mengambil lambang atom pada notasi, sehingga dapat diperoleh panjang senyawa notasi SMILES dan mengetahui jumlah masing-masing elemen. Hasil akhir berupa jumlah masing-masing elemen yang akan dibagi dengan panjang SMILES. Hasil *preprocessing* akan digunakan sebagai *input* dalam proses klastering senyawa aktif menggunakan metode K-Means dengan inisialisasi menggunakan metode *heuristic o(n logn)*. Berikut adalah contoh *preprocessing* notasi SMILES:

Tabel 2.3 *Preprocessing*

Kode SMILES	Fitur											
	Jumlah Atom											Panjang SMILES
	B	C	N	O	P	S	F	Cl	Br	I	OH	
C(CC(C(=O)O)NC(=O)CCC(=O)O)CN=C(N)N	0	10	4	5	0	0	0	0	0	0	0	35

Hasil *preprocessing*:

Tabel 2.4 Hasil *Preprocessing*

B	C	N	O	P	S	F	Cl	Br	I	OH
0	0.33333	0.095238	0.047619	0	0	0	0	0	0	0

2.3 SMILES (*Simplified Molecular Input Line Entry System*)

SMILES merupakan notasi kimia yang dirancang khusus untuk ahli kimia dalam penggunaan komputer. Hal ini cukup fleksibel untuk menginterpretasi notasi kimia independen dan spesifik. Sistem SMILES dirancang agar interaktif dari segi pengguna komputer, ahli kimia maupun sistem itu sendiri. Penggunaan kode SMILES yang sederhana memungkinkan pengguna mengkodekan struktur kimia yang mudah digunakan (Weininger, 1987).

Tata cara penulisan notasi kode SMILES yaitu (Junaedi, 2011):

1. Penulisan Atom

   Penulisan atom disesuaikan dengan simbol atomik senyawa. Penulisan atom dilakukan dengan cara menuliskan huruf besar. Apabila memiliki simbol lebih dari satu huruf, maka huruf pertama huruf besar dan diikuti dengan huruf kecil.

2. Penulisan Ikatan

   Ikatan antar atom terbagi menjadi tiga macam, yang pertama yaitu ikatan tunggal dilambangkan dengan notasi "-", yang kedua yaitu ikatan rangkap yang dilambangkan dengan notasi "=" dan yang ketiga yaitu ikatan rangkap tiga yang dilambangkan dengan notasi "#".

3. Penulisan Percabangan

   Penulisan notasi pada percabangan ditandai dengan kurung buka dan kurung tutup "()".

Notasi SMILES terdiri dari atom yang menyusun suatu senyawa. Atom penyusun notasi SMILES disebut sebagai *organic subset* atau atom kimia organic, yaitu atom yang ditemukan pada organisme lain. Adapun atom-atom yang termasuk dalam *organic subset* menurut Weininger (1988) adalah B (Boron), C (Karbon), N (Nitrogen), O (Oksigen), P (Fosfor), S (Sulfur/Belerang), F (Fluor), Cl (Klorin), Br (Bromin), I (Yodium).

2.4 K-Means

Analisa klaster merupakan kegiatan yang menganalisa kumpulan obyek untuk menemukan kesamaan dan perbedaan sehingga membentuk suatu klaster yang sama maupun berbeda dengan obyek tersebut (Hermawati, 2013). Pengklasteran bertujuan untuk mengelompokkan dan memahami struktur data. Klasterisasi hanya tahap awal untuk kemudian dilanjutkan dengan pengolahan inti dan

pelabelan kelas pada tiap kelompok. Hal ini nantinya dapat digunakan sebagai data latih.

Algoritme klastering K-Means dapat membagi data berdasarkan jarak antar data pada kelompok yang telah ditetapkan. Algoritme ini bergantung pada fungsi untuk mengukur data yang mempunyai ciri khas sama. Jarak itu sendiri dihitung menggunakan fungsi *euclidean.* Kemudian data dimasukkan dalam kelompok yang mempunyai jarak terdekat (Santosa, 2007).

Langkah-langkah pengelompokan data adalah (Santosa, 2007):

1. Pilih jumlah klaster.
2. Inisialisasi awal dan pusat klaster dilakukan secara *random*.
3. Setiap data ditempatkan ke pusat klaster terdekat berdasarkan jarak antar obyek. Pada tahap ini jarak dihitung dengan menentukan kemiripan atau ketidakmiripan data dengan metode jarak *Euclidean* (*Euclidean Distance*) dengan rumus seperti pada persamaan 2.1:

$$d(x,y) = |x-y|^2 = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (2.1)$$

dengan :

$d(x,y)$ = jarak antara *x* dan *y*

$x_i = (x_1, x_2, \dots, x_i)$ yaitu variabel data

$y_i = (y_1, y_2, \dots, y_j)$ yaitu variabel pada titilk pusat

4. Hitung pusat klaster yang baru dengan keanggotaan yang baru dengan cara menghitung rata-rata obyek pada klaster. Penghitungan bisa juga dengan menggunakan median.
5. Hitung kembali jarak tiap objek dengan pusat klaster yang baru, hingga klaster tidak berubah, maka proses pengklasteran selesai.

2.5 *Heuristic O(n logn)*

Algoritme *heuristic n(o logn)* digunakan untuk inisialisasi pusat klaster. Berbeda dengan algoritme K-Means Konvensional yang inisialisasi pusat klasternya ditentukan secara *random*, K-Means dengan *heuristic n (o logn)* pusat klasternya sudah ditentukan di awal dan tidak berubah-ubah. Cara kerja metode ini adalah dengan mempartisi masukan ke sejumlah klaster, kemudian dirata-rata untuk digunakan sebagai nilai awal pusat klaster (Nazeer, 2011). Langkah-langkah metode ini adalah:

1. Pada setiap kolom fitur *dataset* ditentukan nilai terbesar dan terkecil elemen.
2. Menentukan jarak setiap kolom fitur dengan mencari selisih antara nilai terbesar dan terkecil pada poin 1.
3. Data diurutkan dari nilai terkecil ke terbesar berdasarkan nilai jarak terbesar yang sudah dicari pada poin 2.

4. Mempartisi data sejumlah klaster menjadi bagian sama banyak.
5. Menghitung rata-rata tiap fitur untuk masing-masing klaster yang sudah dipartisi pada poin 4.
6. Lakukan perhitungan jarak minimum menggunakan *euclidean distance* berdasarkan pusat klaster awal yang sudah didapat dari poin 5 dengan masing-masing data.

2.6 PHP *Hypertext Processor* (PHP)

Menurut Anhar (2010), PHP singkatan dari PHP Hypertext Preprocessor yaitu bahasa pemrograman web *server-side* yang bersifat *open source*. PHP merupakan kode yang terintegrasi dengan HTML dan berada pada *server* (server side HTML embedded scripting). PHP adalah kode yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru/*up to date*. Semua kode PHP dieksekusi pada *server* yang mana kode tersebut dijalankan. Dapat dikatakan juga bahwa PHP *Hypertext Prepocessor*, yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web.

2.7 Basis Data MySQL

MySQL adalah suatu sistem manajemen data rasional (RDBMS) yang mampu bekerja secara cepat, kokoh dan mudah digunakan (Kadir, 2008). *Database* memungkinkan menyimpan, menelusuri, dan mengurutkan data secara efisien. *Server* MySQL yang membantu melakukan fungsionalitas tersebut. Bahasa yang digunakan MYSQL adalah SQL, standar bahasa *database* yang rasional di seluruh dunia saat ini.

BAB 3 METODOLOGI

Metodologi penelitian dilakukan dengan tahap awal studi literatur, kemudian pengumpulan dan pengolahan data menggunakan sistem yang telah dibuat. Kemudian dilakukan analisa dari metode yang diterapkan pada objek dan hasil akan dituliskan pada bab kesimpulan dan saran. Diagram alir pengerjaan metodologi ditunjukkan pada Gambar 3.1

Gambar 3.1 Diagram Alir Metode Penelitian

3.1 Studi Pustaka

Studi literatur dilakukan untuk melakukan pembelajaran dari literatur berbagai bidang yang berhubungan dengan studi kasus yang dipelajari, meliputi :

1. K-Means
2. *Heuristic o(n logn)*
3. Fungsi Senyawa SMILES

3.2 Pengumpulan Data

Data yang digunakan dalam pengaplikasian metode K-Means dengan inisialisasi klaster menggunakan metode *heuristic o(n logn)* diambil dari situs

pubchem, dimana senyawa yang digunakan adalah senyawa aktif dengan fungsi farmakologi. Data yang dikomputasi adalah senyawa yang telah dikonversi pada kode SMILES.

3.3 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk menganalisa metode K-Means dengan inisialisasi klaster menggunakan metode *heuristic o(n logn)* yang akan digunakan sebagai parameter masukan pada sistem. Kebutuhan yang harus dipenuhi pada program meliputi masukan berupa notasi SMILES, kemudian dilakukan tahap *preprocessing,* inisialisasi klaster dengan metode *heuristic o(n logn),* K-Means klastering dan hasil keluaran berupa pengelompokan jenis obat.

Sistem yang dibangun memiliki beberapa kebutuhan, diantaranya kebutuhan perangkat keras, kebutuhan perangkat lunak dan kebutuhan data. Sistem harus memenuhi kebutuhan yang diperlukan agar dapat berjalan dengan baik.

3.4 Perancangan Sistem

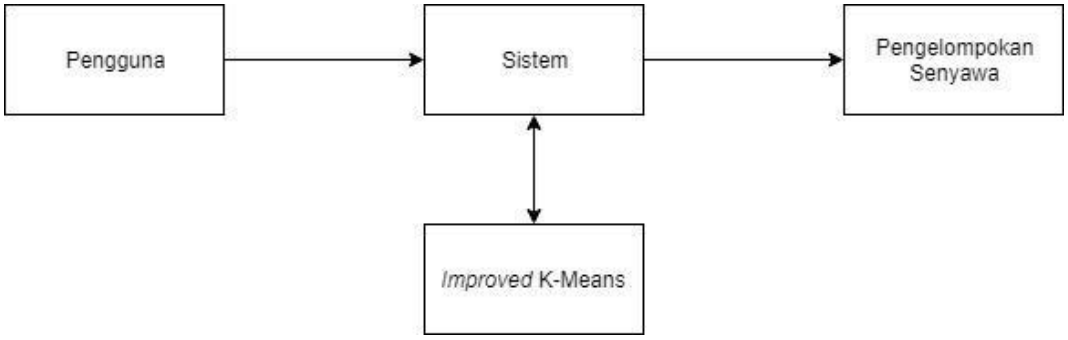
Perancangan dilakukan untuk memudahkan proses implementasi. Perancangan merupakan tahapan implementasi dari teori yang telah dipelajari dengan menggunakan data serta algoritme yang didapatkan untuk merancang sistem klasifikasi senyawa aktif kode SMILES. Tahapan perancangan meliputi diagram alir, manualisasi dan perancangan antarmuka.

3.4.1 Deskripsi Umum Sistem

Sistem pengelompokan/klastering senyawa aktif pada kode SMILES bekerja dengan mengelompokkan fungsi farmakologi dari senyawa tersebut. Penglompokan ini menggunakan metode K-Means dengan inisialisasi klaster menggunakan metode *heuristic o(n logn)*. Masukan berupa fitur dari senyawa kode SMILES meliputi panjang kode SMILES, jumlah masing-masing elemen, dll. Kemudian hasil masukan akan diproses oleh sistem dengan *heuristic n(o logn)* untuk digunakan dalam metode K-Means. Sistem akan menghasilkan *output* berupa pengelompokan senyawa jenis penyakit tertentu.

3.4.2 Cara Kerja Sistem

Perancangan alur kerja sistem ditunjukkan pada Gambar 3.2. Pada gambar dijelaskan bahwa pengguna berinteraksi dengan sistem berbasis web, yaitu dengan memberikan masukan berupa kode SMILES. Sistem akan memproses dimulai dengan melakukan *preprocessing* terlebih dahulu, kemudian sistem menerapkan metode K-Means dengan inisialisasi menggunakan *heuristic o(n logn)*. Sistem menghasilkan pengelompokan senyawa kode SMILES berdasarkan jenis obat suatu penyakit tertentu.



Gambar 3.2 Alur Kerja Sistem

3.5 Implementasi

Implementasi dibuat berdasarkan rancangan yang telah dibuat sebelumnya. Pada tahap ini akan dilakukan implementasi sistem klastering senyawa kode SMILES dengan metode K-Means dengan inisialisasi pusat klaster menggunakan metode *heuristic o(n logn)*. Sistem menggunakan implementasi berbasis web dengan bahasa pemrograman PHP, antarmuka dan juga penyimpanan data.

3.6 Pengujian dan Analisis

Pada tahap pengujian dan analisis, penulis melakukan pengujian dari implementasi yang telah dilakukan sebelumnya. Pengujian dilakukan dengan tujuan mengukur tingkat keberhasilan dari metode yang diterapkan pada permasalahan. Pengujian dilakukan dengan menggunakan 3 metode meliputi:

1. Pengujian Validitas Program
2. Pengujian Data Latih dan Data Uji
3. Pengujian *K-Fold Cross Validation*

Penulis melakukan analisis untuk mengetahui tingkat akurasi sistem dalam mengklaster kode SMILES. Tingkat akurasi dihitung dari keberhasilan data yang diuji dari seluruh data yang ada dengan rumus seperti pada persamaan 3.1:

$$Akurasi = \frac{data\ berhasil}{seluruh\ data} * 100\% \qquad (3.1)$$

Pengujian dilakukan dengan tujuan mengetahui kesalahan yang terjadi pada sistem dan mengetahui kekurangan yang ada pada sistem.

3.7 Kesimpulan dan Saran

Pada tahap kesimpulan dan saran, penulis menarik kesimpulan yang didapat dari perancangan, implementasi dan pengujian yang telah dilakukan. Penulis juga memberikan saran yang dapat dijadikan masukan untuk penelitian selanjutnya.

3.8 Tahapan Penelitian

Adapun tahapan penelitian yang akan dilakukan oleh penulis dalam penelitian kali ini meliputi:

1. Pengumpulan data melalui *website pubchem.ncbi.nlm.nih.gov*. Data yang dikumpulkan akan digunakan sebagai data latih dan data uji.
2. Analisis kebutuhan sistem dilakukan untuk mencari tahu sistem apa yang akan digunakan dan diterapkan.
3. Perancangan sistem dilakukan untuk mengetahui bagaimana sistem akan berjalan. Tahap ini direpresentasikan melalui diagram alir berdasarkan analisis kebutuhan.
4. Implementasi sistem dilakukan berdasarkan perancangan yang telah dibuat dan menggunakan bahasa pemrograman PHP.
5. Pengujian dilakukan untuk mengecek sistem apakah sudah berjalan sesuai dengan kebutuhan dan perancangan yang dibuat.
6. Penyusunan laporan berisi rincian laporan penelitian yang meliputi pendahuluan, landasan kepustakaan, metode penelitian, perancangan, implementasi, hingga kesimpulan dan saran.

BAB 4 PERANCANGAN

4.1 Deskripsi Umum Sistem

Sistem yang akan dibuat betujuan untuk mengetahui penerapan metode K-Means dengan inisialisasi pusat klasternya menggunakan metode *heuristic o (n logn).* Fitur yang digunakan sebagai masukan ada 11, yaitu jumlah masing-masing elemen atom yang kemudian akan dibagi dengan panjang kode SMILES. Sistem akan mengolah masukan dengan cara pembelajaran. SIstem ini mempunyai dua proses, yaitu proses pelatihan dan proses pengujian. Pada proses pelatihan dan pengujian masing-masing memerlukan masukan berupa data latih dan data uji. Proses yang akan dijalankan oleh sistem adalah:

1. Proses Pelatihan
   Proses ini bertujuan untuk mendapatkan nilai pusat klaster dari metode K-Means konvensional dan *improved* K-Means. Nilai pusat klaster didapat nantinya akan digunakan pada proses pengujian dengan data uji. Data latih digunakan untuk mendapatkan nilai pusat klaster yang berjumlah dua klaster.
2. Proses Pengujian
   Proses pengujian menggunakan nilai pusat klaster yang sudah didapatkan dari proses pelatihan yang akan dilakukan proses pengelompokan dengan data uji. Keluaran yang dihasilkan berupa data yang sudah dikelompokkan berdasarkan 2 klaster penyakit.

4.2 Perancangan Sistem

Perancangan dibuat untuk mengetahui bagaimana sistem dengan metode K-Means dengan inisialisasi pusat klaster *heuristic o(n logn)* bekerja pada data senyawa aktif SMILES. Metode *heuristic o(n logn)* digunakan untuk menginisialisasi pusat klaster awal, selanjutnya algoritme K-Means akan memproses pengelompokan data. Alur perancangan sistem ditunjukkan pada Gambar 4.1.

4.2.1 Basis Pengetahuan

Basis pengetahuan berisi pengetahuan yang digunakan untuk memahami, merumuskan dan memecahkan masalah. Basis pengetahuan merupakan representasi pengetahuan dari hasil analisis data SMILES. Terdapat 11 fitur masukan yang digunakan sebagai perhitungan pengelompokan senyawa kode SMILES meliputi jumlah masing-masing elemen B, C, N, O, P, S, F, Cl, Br, I dan OH. Masing-masing fitur akan dibagi dengan panjang senyawa kode SMILES sebelum diproses.

4.2.2 Inisial Pusat Klaster

Metode *heuristic o(n logn)* bekerja dengan membagi data yang sudah diurutkan sebanyak ‘k’ set klaster sama banyak. Kemudian dicari nilai rata-rata tiap klaster, hasil rata-rata akan digunakan sebagai pusat awal klaster (Nazeer, 2011). Alur kerja metode *heuristic o(n logn)* dapat dilihat pada Gambar 4.2.

Perhitungan *Heuristic o(n logn)* pada Gambar 4.2:

1. Untuk setiap kolom data dicari jarak antara nilai minimum dan maksimum, lalu tentukan jarak terbesar
2. Seluruh data diurutkan secara meningkat berdasarkan kolom yang memiliki rentang jarak terbesar
3. Data dipartisi menjadi *K* bagian sama banyak
4. Masing-masing bagian yang telah dipartisi dihitung rata-rata tiap elemennya untuk ditetapkan sebagai nilai awal pusat klaster
5. Lakukan tahap berikut secara berulang:
   a. Menentukan jarak data dengan pusat klaster dan memilih klaster dengan jarak terdekat
   b. Menghitung pusat klaster baru dengan menghitung rata-rata data tiap klaster hingga pusat klaster sudah konvergen

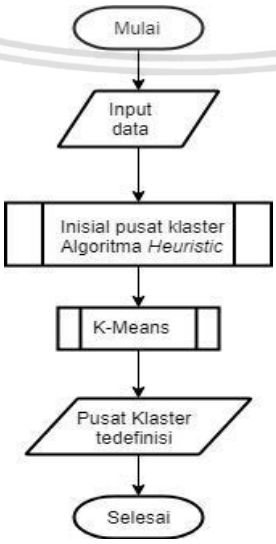
4.2.3 Klasterisasi dengan Algoritme K-Means

Algoritme K-Means disebut juga *portioning clustering* dimana dilakukan proses pemisahan data sejumlah *K* bagian. K-Means dikenal mudah dalam mengklaster data dalam jumlah besar dengan efisiensi waktu yang cepat. Pada penelitian ini proses pengelompokan dengan K-Means membentuk dua buah klaster. Proses K-Means dijabarkan pada Gambar 4.3.

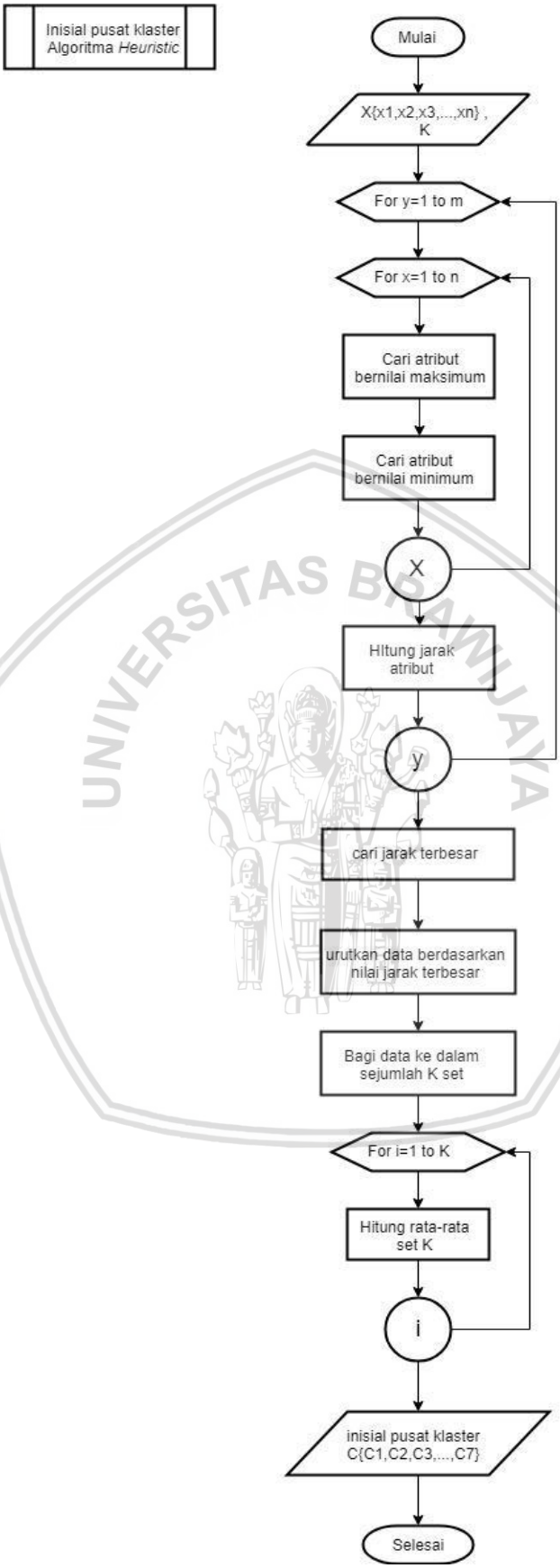
Perhitungan K-Means :

1. Masukkan *dataset* dan pusat klaster
2. Masukan data pada klaster yang mempunyai jarak terdekat tiap data dengan K pusat klaster
3. Menghitung titik pusat klaster baru sejumlah *K*
4. Lakukan perulangan untuk:
   a. Mencari jarak terdekat setiap data dengaan pusat klaster baru
   b. Masukkan data pada klaster jarak terdekat hingga klaster data sudah konvergen

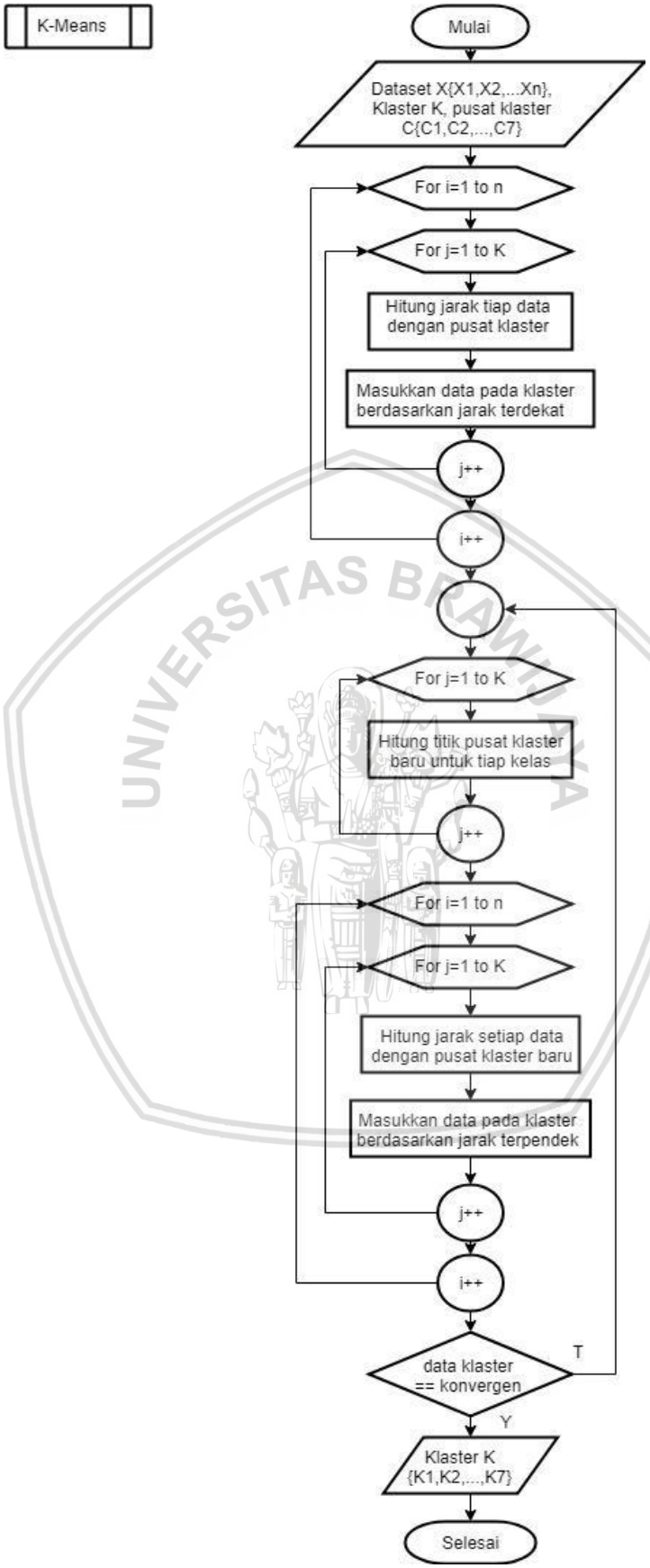
Proses klasterisasi K-Means digambarkan pada Gambar 4.3.



Gambar 4.1 Alur Perancangan Sistem



Gambar 4.2 Algoritme *Heuristic o(n logn)*



Gambar 4.3 Alur Pengklasteran K-Means

4.2.4 Perhitungan Manual

Pada proses perhitungan manual, data latih yang digunakan berjumlah 20 (10 kanker dan 10 metabolisme). Fitur yang digunakan sebanyak 11 yaitu elemen B,C,N,O,P,S,F,Cl,Br,I dan OH. Pelatihan akan diproses menggunakan algoritme K-Means dengan inisialisasi menggunakan algoritme *Heuristic o(n logn)* yaitu pada Tabel 4.1.

Tabel 4.1 Contoh Tabel Data Latih

No	B	C	N	O	P	S	F	Cl	Br	I	OH	Klasifikasi
1	0	0.33333	0.095238	0.047619	0	0	0	0	0	0	0	1
2	0	0.28571	0.114286	0.085714	0	0	0	0	0	0	0.057	1
3	0	0.375	0.041667	0.083333	0	0	0	0	0	0	0.042	1
4	0	0.5625	0.1875	0.0625	0	0	0	0	0	0	0	1
5	0	0.42857	0	0.142857	0	0	0	0	0	0	0.143	1
6	0	0.25	0	0.125	0	0	0	0	0	0	0.125	1
7	0	0.13333	0.066667	0.133333	0	0.0667	0	0	0	0	0.067	1
8	0	0.52941	0.117647	0	0	0	0	0	0	0	0	1
9	0	0.2	0.066667	0.066667	0	0	0	0	0	0	0	1
10	0	0.48276	0.034483	0.034483	0	0	0	0.069	0	0	0.034	1
11	0	0.44737	0.026316	0	0	0	0	0	0	0	0.105	2
12	0	0.55556	0.074074	0.037037	0	0	0	0	0	0	0.037	2
13	0	0.45238	0.047619	0.047619	0	0	0	0	0	0	0.048	2
14	0	0.58696	0.086957	0.021739	0	0	0	0	0	0	0	2
15	0	0.4	0.066667	0.033333	0	0	0	0	0	0	0.067	2
16	0	0.44444	0.055556	0.027778	0	0	0	0	0	0	0.056	2
17	0	0.56818	0.022727	0.022727	0	0	0	0	0	0	0.045	2
18	0	0.28571	0.035714	0.071429	0	0	0	0.0357	0	0	0	2
19	0	0.39623	0.018868	0.075472	0	0.0189	0	0	0	0	0	2
20	0	0.54839	0.032258	0.096774	0	0	0	0	0	0	0	2

Setiap kriteria mewakili nilai elemen yang telah dibagi dengan panjang kode SMILES. Sedangkan kelompok kanker direpresentasikan angka 1, untuk metabolisme angka 2. Langkah awal dilakukan pembangkitan pusat klaster yang dilakukan secara acak. Dalam data latih yang dipilih sebagai pusat klaster yaitu data 1,10 dan 20. Pada K-Means konvensional pusat klaster awal dipilih secara *random* sebanyak K klaster dari data latih. Sedangkan *improved* K-Means menggunakan metode *heuristic o(n logn).* Perhitungan awal pusat klaster awal dengan metode *heuristic o(n logn)* ditunjukkan pada Tabel 4.2.

Tabel 4.2 Jarak Tiap Fitur

No	B	C	N	O	P	S	F	Cl	Br	I	OH	Klasifikasi
1	0	0.33333	0.095238	0.047619	0	0	0	0	0	0	0	1
2	0	0.28571	0.114286	0.085714	0	0	0	0	0	0	0.057	1
3	0	0.375	0.041667	0.083333	0	0	0	0	0	0	0.042	1

Tabel 4.2 (Lanjutan)

No	B	C	N	O	P	S	F	Cl	Br	I	OH	Klasifikasi
4	0	0.5625	0.1875	0.0625	0	0	0	0	0	0	0	1
5	0	0.42857	0	0.142857	0	0	0	0	0	0	0.143	1
6	0	0.25	0	0.125	0	0	0	0	0	0	0.125	1
7	0	0.13333	0.066667	0.133333	0	0.0667	0	0	0	0	0.067	1
8	0	0.52941	0.117647	0	0	0	0	0	0	0	0	1
9	0	0.2	0.066667	0.066667	0	0	0	0	0	0	0	1
10	0	0.48276	0.034483	0.034483	0	0	0	0.069	0	0	0.034	1
11	0	0.44737	0.026316	0	0	0	0	0	0	0	0.105	2
12	0	0.55556	0.074074	0.037037	0	0	0	0	0	0	0.037	2
13	0	0.45238	0.047619	0.047619	0	0	0	0	0	0	0.048	2
14	0	0.58696	0.086957	0.021739	0	0	0	0	0	0	0	2
15	0	0.4	0.066667	0.033333	0	0	0	0	0	0	0.067	2
16	0	0.44444	0.055556	0.027778	0	0	0	0	0	0	0.056	2
17	0	0.56818	0.022727	0.022727	0	0	0	0	0	0	0.045	2
18	0	0.28571	0.035714	0.071429	0	0	0	0.0357	0	0	0	2
19	0	0.39623	0.018868	0.075472	0	0.0189	0	0	0	0	0	2
20	0	0.54839	0.032258	0.096774	0	0	0	0	0	0	0	2
Min	**0**	**0.13333**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	
Max	**0**	**0.58696**	**0.1875**	**0.142857**	**0**	**0.0667**	**0**	**0.069**	**0**	**0**	**0.143**	
Jarak	**0**	**0.45362**	**0.1875**	**0.142857**	**0**	**0.0667**	**0**	**0.069**	**0**	**0**	**0.143**	

Identifikasi kolom yang memiliki jarak terbesar, pada data latih diketahui jarak terbesar terdapat pada kolom C. Lakukan *sorting*/pengurutan data dari yang terkecil hingga terbesar berdasarkan data pada kolom C, sedangkan fitur yang lain menyesuaikan. Contoh hasil pengurutan dapat dilihat pada Tabel 4.3.

Tabel 4.3 Pengurutan Data Berdasarkan Kolom Jarak Terbesar

No	B	C	N	O	P	S	F	Cl	Br	I	OH	Klasifikasi
1	0	0.13333	0.066667	0.133333	0	0.0667	0	0	0	0	0.067	1
2	0	0.2	0.066667	0.066667	0	0	0	0	0	0	0	1
3	0	0.25	0	0.125	0	0	0	0	0	0	0.125	1
4	0	0.28571	0.114286	0.085714	0	0	0	0	0	0	0.057	1
5	0	0.28571	0.035714	0.071429	0	0	0	0.0357	0	0	0	2
6	0	0.33333	0.095238	0.047619	0	0	0	0	0	0	0	1
7	0	0.375	0.041667	0.083333	0	0	0	0	0	0	0.042	1
8	0	0.39623	0.018868	0.075472	0	0.0189	0	0	0	0	0	2
9	0	0.4	0.066667	0.033333	0	0	0	0	0	0	0.067	2
10	0	0.42857	0	0.142857	0	0	0	0	0	0	0.143	1
11	0	0.44444	0.055556	0.027778	0	0	0	0	0	0	0.056	2
12	0	0.44737	0.026316	0	0	0	0	0	0	0	0.105	2

Tabel 4.3 (Lanjutan)

No	B	C	N	O	P	S	F	Cl	Br	I	OH	Klasifikasi
13	0	0.45238	0.047619	0.047619	0	0	0	0	0	0	0.048	2
14	0	0.48276	0.034483	0.034483	0	0	0	0.069	0	0	0.034	1
15	0	0.52941	0.117647	0	0	0	0	0	0	0	0	1
16	0	0.54839	0.032258	0.096774	0	0	0	0	0	0	0	2
17	0	0.55556	0.074074	0.037037	0	0	0	0	0	0	0.037	2
18	0	0.5625	0.1875	0.0625	0	0	0	0	0	0	0	1
19	0	0.56818	0.022727	0.022727	0	0	0	0	0	0	0.045	2
20	0	0.58696	0.086957	0.021739	0	0	0	0	0	0	0	2
Min	**0**	**0.13333**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	
Max	**0**	**0.58696**	**0.1875**	**0.142857**	**0**	**0.0667**	**0**	**0.069**	**0**	**0**	**0.143**	
Jarak	**0**	**0.45362**	**0.1875**	**0.142857**	**0**	**0.0667**	**0**	**0.069**	**0**	**0**	**0.143**	

Setelah data diurutkan berdasarkan kolom dengan jarak terbesar yaitu kolom C, data dipartisi sejumlah K bagian yaitu 1,2 sama banyak secara terurut pada kolom klasifikasi. Contoh partisi data dapat dilihat pada Tabel 4.4.

Tabel 4.4 Partisi Data

No	B	C	N	O	P	S	F	Cl	Br	I	OH	Klasifikasi
1	0	0.13333	0.066667	0.133333	0	0.0667	0	0	0	0	0.067	1
2	0	0.2	0.066667	0.066667	0	0	0	0	0	0	0	1
3	0	0.25	0	0.125	0	0	0	0	0	0	0.125	1
4	0	0.28571	0.114286	0.085714	0	0	0	0	0	0	0.057	1
5	0	0.28571	0.035714	0.071429	0	0	0	0.0357	0	0	0	1
6	0	0.33333	0.095238	0.047619	0	0	0	0	0	0	0	1
7	0	0.375	0.041667	0.083333	0	0	0	0	0	0	0.042	1
8	0	0.39623	0.018868	0.075472	0	0.0189	0	0	0	0	0	1
9	0	0.4	0.066667	0.033333	0	0	0	0	0	0	0.067	1
10	0	0.42857	0	0.142857	0	0	0	0	0	0	0.143	1
11	0	0.44444	0.055556	0.027778	0	0	0	0	0	0	0.056	2
12	0	0.44737	0.026316	0	0	0	0	0	0	0	0.105	2
13	0	0.45238	0.047619	0.047619	0	0	0	0	0	0	0.048	2
14	0	0.48276	0.034483	0.034483	0	0	0	0.069	0	0	0.034	2
15	0	0.52941	0.117647	0	0	0	0	0	0	0	0	2
16	0	0.54839	0.032258	0.096774	0	0	0	0	0	0	0	2
17	0	0.55556	0.074074	0.037037	0	0	0	0	0	0	0.037	2
18	0	0.5625	0.1875	0.0625	0	0	0	0	0	0	0	2
19	0	0.56818	0.022727	0.022727	0	0	0	0	0	0	0.045	2
20	0	0.58696	0.086957	0.021739	0	0	0	0	0	0	0	2
Min	**0**	**0.13333**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	
Max	**0**	**0.58696**	**0.1875**	**0.142857**	**0**	**0.0667**	**0**	**0.069**	**0**	**0**	**0.143**	
Jarak	**0**	**0.45362**	**0.1875**	**0.142857**	**0**	**0.0667**	**0**	**0.069**	**0**	**0**	**0.143**	

Setelah data dibagi sejumlah K kelas sama banyak, hitung rata-rata masing-masing atribut tiap kelas. Hasil perhitungan rata-rata digunakan sebagai pusat klaster awal yaitu C1 dan C2. Contoh perhitungan rata-rata tiap kelas dan pusat klaster awal dapat dilihat pada Tabel 4.5 dan Tabel 4.6.

Tabel 4.5 Perhitungan Rata-Rata Setiap Kelas

No	B	C	N	O	P	S	F	Cl	Br	I	OH	Klasifikasi
1	0	0.13333	0.066667	0.133333	0	0.0667	0	0	0	0	0.067	1
2	0	0.2	0.066667	0.066667	0	0	0	0	0	0	0	1
3	0	0.25	0	0.125	0	0	0	0	0	0	0.125	1
4	0	0.28571	0.114286	0.085714	0	0	0	0	0	0	0.057	1
5	0	0.28571	0.035714	0.071429	0	0	0	0.0357	0	0	0	1
6	0	0.33333	0.095238	0.047619	0	0	0	0	0	0	0	1
7	0	0.375	0.041667	0.083333	0	0	0	0	0	0	0.042	1
8	0	0.39623	0.018868	0.075472	0	0.0189	0	0	0	0	0	1
9	0	0.4	0.066667	0.033333	0	0	0	0	0	0	0.067	1
10	0	0.42857	0	0.142857	0	0	0	0	0	0	0.143	1
Rata-Rata	**0**	**0.30879**	**0.050577**	**0.086476**	**0**	**0.0086**	**0**	**0.0036**	**0**	**0**	**0.05**	
11	0	0.44444	0.055556	0.027778	0	0	0	0	0	0	0.056	2
12	0	0.44737	0.026316	0	0	0	0	0	0	0	0.105	2
13	0	0.45238	0.047619	0.047619	0	0	0	0	0	0	0.048	2
14	0	0.48276	0.034483	0.034483	0	0	0	0.069	0	0	0.034	2
15	0	0.52941	0.117647	0	0	0	0	0	0	0	0	2
16	0	0.54839	0.032258	0.096774	0	0	0	0	0	0	0	2
17	0	0.55556	0.074074	0.037037	0	0	0	0	0	0	0.037	2
18	0	0.5625	0.1875	0.0625	0	0	0	0	0	0	0	2
19	0	0.56818	0.022727	0.022727	0	0	0	0	0	0	0.045	2
20	0	0.58696	0.086957	0.021739	0	0	0	0	0	0	0	2
Rata-Rata	**0**	**0.51779**	**0.068514**	**0.035066**	**0**	**0**	**0**	**0.0069**	**0**	**0**	**0.033**	

Tabel 4.6 Pusat Klaster Awal

Centroid / Pusat Klaster	B	C	N	O	P	S	F	Cl	Br	I	OH
C1	0	0.30879	0.050577	0.086476	0	0.0086	0	0.0036	0	0	0.05
C2	0	0.51779	0.068514	0.035066	0	0	0	0.0069	0	0	0.033

Kemudian dilakukan perhitungan jarak masing-masing data pada Tabel 4.1 dengan pusat klaster pada Tabel 4.6 dengan rumus *euclidean distance*. Contoh perhitungan data 1 dengan klaster 1 dan klaster 2:

$$d(x1, C1)$$
$$= \sqrt{\begin{matrix}(0-0)^2 + (0.30879 - 0.3333)^2 + (0.050577 - 0.095)^2 + (0.086476 - 0.0476)^2 \\ +(0-0)^2 + (0.0086 - 0)^2 + (0-0)^2 + (0.0036 - 0)^2 + (0-0)^2 + (0-0)^2 \\ +(0.05 - 0)^2\end{matrix}}$$
$$= 0.0818093$$

$$d(x1, C2)$$
$$= \sqrt{\begin{matrix}(0-0)^2 + (0.51779 - 0.3333)^2 + (0.068514 - 0.095)^2 + (0.035066 - 0.0476)^2 \\ +(0-0)^2 + (0-0)^2 + (0-0)^2 + (0.0069 - 0)^2 + (0-0)^2 + (0-0)^2 \\ +(0.033 - 0)^2\end{matrix}}$$
$$= 0.18975$$

Lakukan perhitungan hingga seluruh data pada Tabel 4.1.

Hasil perhitungan jarak masing-masing data dengan pusat klaster menggunakan *eucledean distance* pada iterasi ke-1 ditunjukkan pada Tabel 4.7.

Tabel 4.7 Hasil Perhitungan *Euclidean Distance* iterasi 1

Data No	C1	C2	Terdekat	Klaster
1	0.0818093	0.18975	0.081809	1
2	0.0687658	0.24326	0.068766	1
3	0.068033	0.15353	0.068033	1
4	0.2937307	0.13422	0.134222	2
5	0.1696851	0.19103	0.169685	1
6	0.1149316	0.30511	0.114932	1
7	0.1921119	0.40389	0.192112	1
8	0.2514689	0.06989	0.069894	2
9	0.1227693	0.32109	0.122769	1
10	0.1944684	0.07901	0.079008	2
11	0.1743869	0.11535	0.115352	2
12	0.2532663	0.0391	0.039096	2
13	0.1490931	0.07175	0.07175	2
14	0.2923648	0.08005	0.080048	2
15	0.1084715	0.12286	0.108472	1
16	0.148288	0.0786	0.078603	2
17	0.2687574	0.07072	0.070724	2
18	0.0677208	0.24114	0.067721	1
19	0.1067285	0.14261	0.106728	1
20	0.2458346	0.08465	0.084645	2

Penentuan keanggotaan klaster ditentukan berdasarkan jarak terkecil. Misal hasil pencarian jarak data x_1 terhadap pusat klaster 1 dan 2 kemudian mencari jarak terkecil dimana terletak pada klaster 1, maka data x_1 masuk anggota klaster 1 (C1). Hasil perhitungan jarak data dengan pusat klaster pada Tabel 4.8 diperoleh hasil dengan

anggota klaster 1 sebanyak 10 data yaitu $x_1, x_2, x_3, x_5, x_6, x_7, x_9, x_{15}, x_{18}, x_{19}$. Sedangkan anggota klaster 2 (C2) sebanyak 10 data yaitu $x_4, x_8, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{16}, x_{17}, x_{20}$.

Nilai pusat klaster baru didapatkan dengan cara menghitung rata-rata (*mean*) dari data anggota tiap klaster. Sehingga masing-masing klaster mendapatkan nilai baru untuk tiap fitur. Tabel hasil rata-rata data tiap klaster dapat dilihat pada Tabel 4.8.

Tabel 4.8 Rata-Rata Setiap Klaster

No	B	C	N	O	P	S	F	Cl	Br	I	OH	Klasifikasi
KLASTER 1												
1	0	0.33333	0.095238	0.047619	0	0	0	0	0	0	0	1
2	0	0.28571	0.114286	0.085714	0	0	0	0	0	0	0.057	1
3	0	0.375	0.041667	0.083333	0	0	0	0	0	0	0.042	1
5	0	0.42857	0	0.142857	0	0	0	0	0	0	0.143	1
6	0	0.25	0	0.125	0	0	0	0	0	0	0.125	1
7	0	0.13333	0.066667	0.133333	0	0.0667	0	0	0	0	0.067	1
9	0	0.2	0.066667	0.066667	0	0	0	0	0	0	0	1
15	0	0.4	0.066667	0.033333	0	0	0	0	0	0	0.067	1
18	0	0.28571	0.035714	0.071429	0	0	0	0.0357	0	0	0	1
19	0	0.39623	0.018868	0.075472	0	0.0189	0	0	0	0	0	1
RataRata	**0**	**0.30879**	**0.050577**	**0.086476**	**0**	**0.0086**	**0**	**0.0036**	**0**	**0**	**0.05**	
KLASTER 2												
4	0	0.5625	0.1875	0.0625	0	0	0	0	0	0	0	2
8	0	0.52941	0.117647	0	0	0	0	0	0	0	0	2
10	0	0.48276	0.034483	0.034483	0	0	0	0.069	0	0	0.034	2
11	0	0.44737	0.026316	0	0	0	0	0	0	0	0.105	2
12	0	0.55556	0.074074	0.037037	0	0	0	0	0	0	0.037	2
13	0	0.45238	0.047619	0.047619	0	0	0	0	0	0	0.048	2
14	0	0.58696	0.086957	0.021739	0	0	0	0	0	0	0	2
16	0	0.44444	0.055556	0.027778	0	0	0	0	0	0	0.056	2
17	0	0.56818	0.022727	0.022727	0	0	0	0	0	0	0.045	2
20	0	0.54839	0.032258	0.096774	0	0	0	0	0	0	0	2
RataRata	**0**	**0.51779**	**0.068514**	**0.035066**	**0**	**0**	**0**	**0.0069**	**0**	**0**	**0.033**	

Sehingga hasil perhitungan pusat klaster baru ditunjukkan pada Tabel 4.9.

Tabel 4.9 Pusat Klaster Baru

Pusat Klaster	B	C	N	O	P	S	F	Cl	Br	I	OH
C1	0	0.30879	0.050577	0.086476	0	0.0086	0	0.0036	0	0 0.05	0
C2	0	0.51779	0.068514	0.035066	0	0	0	0.0069	0	0 0.033	0

Pusat klaster baru mempunyai hasil sama dengan yang sebelumnya/sudah konvergen, sehingga iterasi dihentikan. Pusat klaster terakhir digunakan untuk proses

pengujian/*testing.* Data uji yang digunakan sebanyak 10, yaitu 5 data kanker dan 5 data metabolisme. Contoh tabel data untuk pengujian ditunjukkan pada Tabel 4.10.

Tabel 4.10 Tabel Data Uji

No	B	C	N	O	P	S	F	Cl	Br	I	OH	Klasifikasi
1	0	0.2381	0.047619	0.095238	0	0	0	0	0	0	0	1
2	0	0.33333	0.095238	0.095238	0.047619	0	0	0.0952	0	0	0	1
3	0	0.52941	0	0	0	0	0	0	0	0	0.059	1
4	0	0.2	0	0	0	0	0	0.3	0	0	0	1
5	0	0.60417	0	0.020833	0	0	0	0	0	0	0.021	1
6	0	0.33333	0.051282	0	0	0	0.076923	0.0256	0	0	0.026	2
7	0	0.39286	0.035714	0	0	0	0	0.0357	0	0	0.071	2
8	0	0.42424	0	0.151515	0	0	0	0	0	0	0	2
9	0	0.40909	0.045455	0	0	0	0	0	0	0	0.136	2
10	0	0.375	0.041667	0.041667	0	0	0	0.0417	0	0	0.083	2

Menghitung jarak data uji dengan klaster yang telah didapat dari proses pelatihan. Hasil perhitungan jarak menggunakan *euclidean distance* dapat dilihat pada Tabel 4.11.

Tabel 4.11 Hasil Pengujian Dengan *Euclidean Distance*

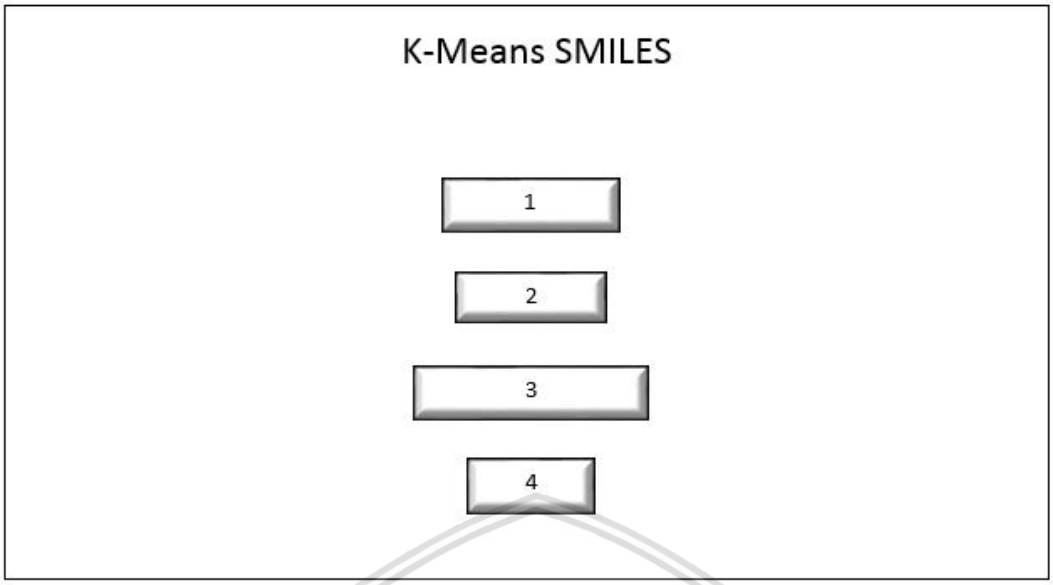
Data No	C1	C2	Klaster	Akurasi
1	0.0875734	0.28878	1	Benar
2	0.1261637	0.22247	1	Benar
3	0.24264	0.08244	2	Salah
4	0.3351332	0.44032	1	Benar
5	0.308304	0.11199	2	Salah
6	0.1230923	0.20462	1	Salah
7	0.1069137	0.13794	1	Salah
8	0.1506769	0.16767	1	Salah
9	0.1584591	0.15622	2	Benar
10	0.0954287	0.15793	1	Salah

Dari 10 data uji yaitu 1-5 data kanker dan 6-10 data metabolisme adalah sebesar 4 data benar. Sehingga akurasi yang didapat dari pengujian adalah sebesar 4/10*100% yaitu 40%.

4.2.5 Perancangan Antarmuka

4.2.5.1 Halaman Awal

Perancangan antarmuka halaman awal sistem diilustrasikan pada Gambar 4.4.



Gambar 4.4 Antarmuka Halaman Awal

Penjelasan:

1. *Button Input* data digunakan untuk menampilkan halaman masukkan data set pelatihan
2. *Button improved* K-Means untuk menampilkan halaman pelatihan *Improved* K-Means
3. *Button* K-Means untuk menampilkan halaman pelatihan K-Means
4. *Button* testing untuk menampilakan halaman pengujian

4.2.5.2 Halaman *Input* data

Perancangan antarmuka halaman *Input* data diilustrasikan pada Gambar 4.5.

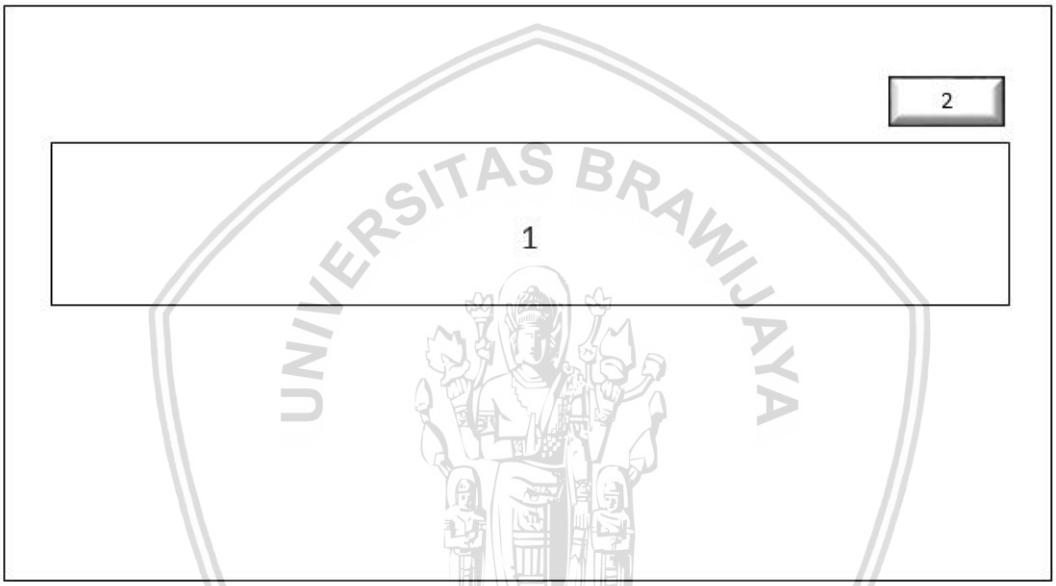
Gambar 4.5 Antarmuka Halaman *Input* data

Penjelasan:

1. *Textbox* untuk memasukkan data berupa senyawa aktif SMILES
2. *Button Input* digunakan untuk memasukkan data yang telah ditulis pada *textbox*
3. Tabel yang berisi tampilan data yang telah dimasukkan ke dalam sistem
4. *Button* menu digunakan untuk menampilkan kembali halaman awal sistem

4.2.5.3 Halaman *Improved* K-Means

Perancangan antarmuka halaman *Improved* K-Means diilustrasikan pada Gambar 4.6.



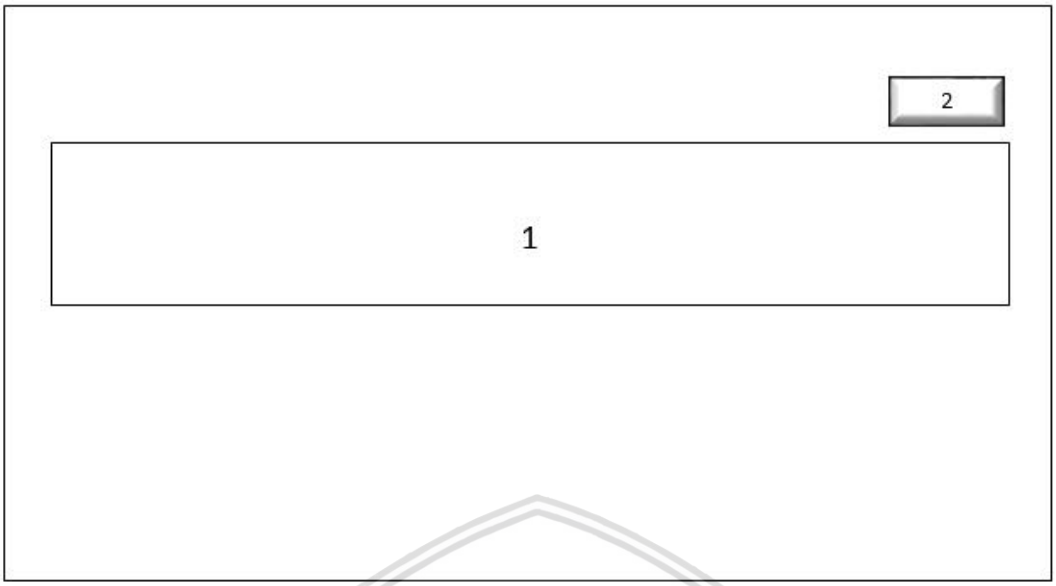
Gambar 4.6 Antarmuka Halaman *Improved* K-Means

Penjelasan:

1. Tabel yang menampilkan hasil pemrosesan dengan metode K-Means *heuristic o (n logn)* yaitu berupa 7 pusat klaster
2. *Button* menu digunakan untuk menampilkan kembali halaman awal sistem

4.2.5.4 Halaman K-Means

Perancangan antarmuka halaman K-Means diilustrasikan pada Gambar 4.7.



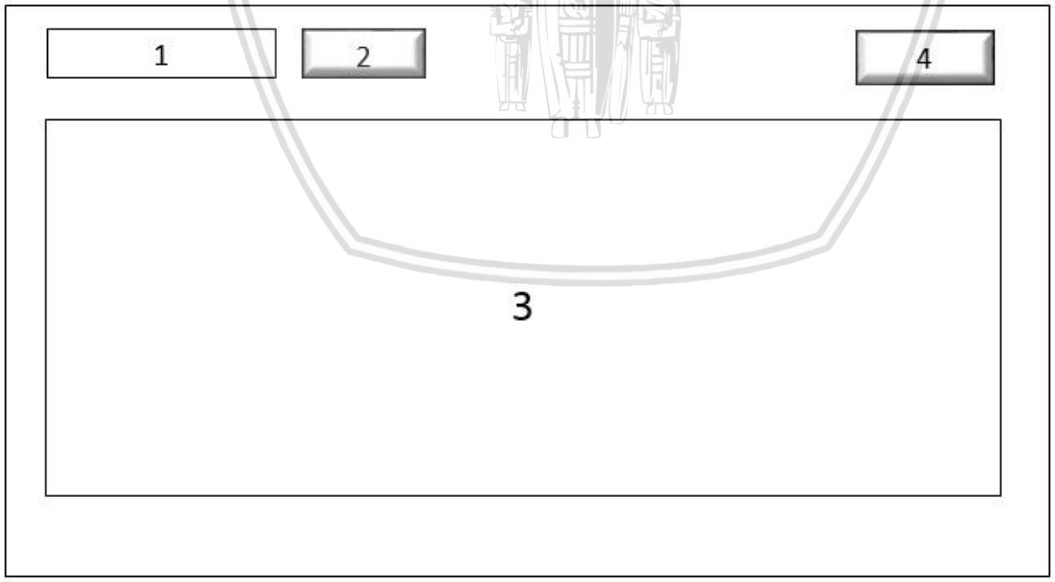
Gambar 4.7 Antarmuka Halaman K-Means

Penjelasan:

1. Tabel yang menampilkan hasil pemrosesan dengan metode K-Means yaitu berupa 7 pusat klaster
2. *Button* menu digunakan untuk menampilkan kembali halaman awal sistem

4.2.5.5 Halaman *Testing*

Perancangan antarmuka halaman *Testing* diilustrasikan pada Gambar 4.8.



Gambar 4.8 Antarmuka Halaman *Testing*

Penjelasan:

1. *Textbox* untuk memasukkan data berupa senyawa aktif SMILES

2. *Button Input* digunakan untuk memasukkan data yang telah ditulis pada *textbox*
3. Tabel yang berisi tampilan data yang telah dimasukkan ke dalam sistem
4. *Button* proses digunakan untuk menampilkan hasil proses pengujian pada halaman hasil *testing*

4.2.5.6 Halaman Hasil *Testing*

Perancangan antarmuka halaman hasil *testing* diilustrasikan pada Gambar 4.9.

Gambar 4.9 Antarmuka Halaman Hasil *Testing*

Penjelasan:

1. Tabel yang menampilkan hasil proses pengujian yaitu berupa hasil akurasi dan jumlah iterasi dari K-Means dan K-Means dengan inisial *heuristic o(n logn)*
2. *Button* menu digunakan untuk menampilkan kembali halaman awal sistem

4.3 Perancangan Pengujian

Perancangan pengujian digunakan pada tahap pengujian guna mengetahui bagaimana hasil dan kualitas sistem yang telah dibuat.

4.3.1 Pengujian Validitas Program

Pengujian Validitas dilakukan untuk menguji program apakah hasil keluaran sudah sesuai dengan perhitungan manual. Pada pengujian ini data yang digunakan pada program sama dengan data pada perhitungan manual. Program dikatakan valid apabila hasil keluaran yang dihasilkan sama dengan hasil dari perhitungan manual.

4.3.2 Rancang Uji Akurasi menggunakan *K-Fold Cross Validation*

Pengujian menggunakan *K-Fold Cross Validation* pada program dimaksudkan untuk mengetahui akurasi yang dilakukan pada sampel data secara acak. Metode ini membagi data uji sejumlah *k dataset* dengan perulangan sebanyak *k* iterasi. Perulangan yang dijalankan menggunakan data latih dan data uji yang berbeda sesuai dengan pembagian sebanyak *k* data. Pengujian pada penelitian ini membagi data sebanyak 10.

BAB 5 HASIL

Bab ini berisi implementasi dan penjelasan tahapan kode program perhitungan dengan metode K-Means dengan inisial pusat klaster *heuristic o(n logn).* Bab ini juga berisi hasil implementasi dan penjelasan antarmuka yang sudah dibuat.

5.1 Lingkungan Implementasi

Lingkungan implementasi membahas tentang komponen yang digunakan dalam pengembangan sistem. Lingkungan implementasi yang digunakan dalam implementasi K-Means dengan inisialisasi pusat klaster menggunakan metode *Heuristic O(N LogN)* meliputi lingkungan perangkat keras dan lingkungan perangkat lunak.

5.1.1 Lingkungan Perangkat Keras

Lingkungan perangkat keras yang digunakan untuk mendukung penelitian ini adalah:

1. Laptop Prosesor Intel(R) Core(TM) i5-2410M CPU @2.30 GHz.
2. Memori RAM 6 GB DDR3.
3. Harddisk 64 0GB.

5.1.2 Lingkungan Perangkat Lunak

Lingkungan perangkat lunak yang digunakan pada penelitian meliputi:

1. Sistem Operasi Windows 8 64bit.
2. XAMPP versi 3.2.2 dengan Apache Web Server untuk implementasi PHP dan MySQL sebagai *database management system* (DBMS).
3. Notepad++ untuk implementasi bahasa pemrograman.
4. Browser Google Chrome versi 67.

5.2 Batasan Implementasi

Agar implementasi algoritme K-Means dengan inisialisasi pusat klaster menggunakan metode *Heuristic O(N LogN)* dapat terfokus dengan baik, maka terdapat batasan implementasi meliputi:

1. *Preprocessing* notasi SMILES dilakukan menggunakan fungsi *regular expression* (*regex*) dimana terdapat pada bahasa pemrograman PHP.
2. Algoritma yang digunakan untuk implementasi program adalah K-Means dan *Heuristic O(N LogN).*
3. Masukan yang digunakan pada program berupa notasi SMILES dan kelas dari data tersebut yang akan digunakan sebagai data latih.

4. Pada proses pelatihan, keluaran yang dihasilkan berupa pusat klaster yang akan digunakan pada proses pengujian.
5. Pada proses pengujian, keluaran yang dihasilkan adalah hasil klasifikasi data uji dan akurasi yang diperoleh.

5.3 Implementasi Program

5.3.1 Proses *Input* Data

Proses *input* data digunakan untuk memasukkan data latih maupun data uji yang akan diproses dengan metode K-Means pada proses pelatihan maupun pengujian. Proses *input* data dapat dilihat pada Source Code 5.1.

```
1.  if(isset($_POST['btnSubmit'])){
2.              $cek_kembar=mysql_num_rows(mysql_query("SELECT *
3.  FROM tbl_senyawa WHERE
4.  senyawa_smiles='".$_POST['txtSenyawa']."'"));
5.
6.              if ($cek_kembar > 0) {
7.                  echo '<script language="javascript">
8.                  alert ("Senyawa Sudah Diinputkan");
9.                  </script>';
10.                 exit();
11.             }
12.
13.             $tampungSenyawa = $_POST['txtSenyawa'];
14.             $C = 0;$B = 0;$S = 0;$N = 0;$P = 0; $F = 0;$I =
15. 0;$O = 0;
16.             $Br = 0;$Cl = 0;$OH = 0;
17.             $panjangString = 0;
18.             $detectAngka =0;
19.
20.             for ($x = 0; $x <= strlen($tampungSenyawa) - 1;
21. $x++) {
22.                 if(substr($tampungSenyawa,$x,2) == "Br"){
23.                     $Br++;
24.                     $x++;
25.                     $detectAngka++;
26.                 }
27.                 elseif(substr($tampungSenyawa,$x,2) ==
28. "Cl"){
29.                     $Cl++;
30.                     $x++;
31.                     $detectAngka++;
32.                 }
33.                 elseif(substr($tampungSenyawa,$x,2) ==
34. "OH"){
35.                     $OH++;
36.                     $x++;
37.                     $detectAngka++;
38.                 }
39.                 elseif(substr($tampungSenyawa,$x,1) == "C"){
40.                     $C++;
41.                 }
42.                 elseif(substr($tampungSenyawa,$x,1) == "B"){
43.                     $B++;
```

```
                    }
                    elseif(substr($tampungSenyawa,$x,1) == "N"){
                          $N++;
                    }
                    elseif(substr($tampungSenyawa,$x,1) == "P"){
                          $P++;
                    }
                    elseif(substr($tampungSenyawa,$x,1) == "F"){
                          $F++;
                    }
                    elseif(substr($tampungSenyawa,$x,1) == "I"){
                          $I++;
                    }
                    elseif(substr($tampungSenyawa,$x,1) == "O"){
                          if
(preg_match('/[\'^£$%&*()}{@#~?><>,|=_+¬-]/',
substr($tampungSenyawa,$x-1,1)) AND
preg_match('/[\'^£$%&*()}{@#~?><>,|=_+¬-]/',
substr($tampungSenyawa,$x+1,1)))
                          {
                                $OH++;

      }elseif(preg_match('/[\'^£$%&*()}{@#~?><>,|=_+¬-]/',
substr($tampungSenyawa,$x-1,1)) AND
is_numeric(substr($tampungSenyawa,$x+1,1)))
                          {

      if(preg_match('/[\'^£$%&*()}{@#~?><>,|=_+¬-]/',
substr($tampungSenyawa,$x+2,1))){
                                $OH++;
                          }else{
                                $O++;
                          }

      }elseif(preg_match('/[\'^£$%&*()}{@#~?><>,|=_+¬-]/',
substr($tampungSenyawa,$x+1,1)) AND
is_numeric(substr($tampungSenyawa,$x-1,1)))
                          {

      if(preg_match('/[\'^£$%&*()}{@#~?><>,|=_+¬-]/',
substr($tampungSenyawa,$x-2,1))){
                                $OH++;
                          }else{
                                $O++;
                          }
                    }else{
                          $O++;
                    }

                    }

                          if
(is_numeric(substr($tampungSenyawa,$x,1)))
                          {
                                $detectAngka++;
                          }

            }

```

```
103.              if
104.  (substr($tampungSenyawa,strlen($tampungSenyawa)-1,1) == "O"){
105.                    $OH++;
106.                    $O--;
107.              }
```

Source Code 5.1 Kode Program Proses *Input* Data

Penjelasan:

Baris 1-11 : Untuk memasukkan data berupa senyawa SMILES dan mengecek supaya tidak ada data masukkan yang sama

Baris 20-93 : Proses *preprocessing*, yaitu memecah senyawa dan menghitung data tiap elemen untuk dilakukan perhitungan selanjutnya

5.3.2 Metode *Heuristic O(N LogN)*

5.3.2.1 Perhitungan Nilai Maksimum, Minimum dan Jarak

Proses selanjutnya pada metode *improved* K-Means yaitu mencari nilai jarak atau selisih dari nilai maksimum dan minimum tiap kolom data. Program mencari nilai maksimum, minimum dan jarak dapat dilihat pada Source Code 5.2.

```
1.  $jarak_b = 0;$jarak_c = 0;$jarak_n = 0;$jarak_o = 0;$jarak_p =
2.  0;$jarak_s = 0;$jarak_f = 0;$jarak_cl = 0;$jarak_br =
3.  0;$jarak_i = 0;$jarak_oh = 0;
4.
5.                    $result= mysql_query("SELECT MIN(
6.  `senyawa_b` ) AS  `lowest_b` , MAX(  `senyawa_b` ) AS
7.  `highest_b`, MIN(  `senyawa_c` ) AS  `lowest_c` , MAX(
8.  `senyawa_c` ) AS  `highest_c`, MIN(  `senyawa_n` ) AS
9.  `lowest_n` , MAX(  `senyawa_n` ) AS  `highest_n`, MIN(
10. `senyawa_o` ) AS  `lowest_o` , MAX(  `senyawa_o` ) AS
11. `highest_o`, MIN(  `senyawa_p` ) AS  `lowest_p` , MAX(
12. `senyawa_p` ) AS  `highest_p`, MIN(  `senyawa_s` ) AS
13. `lowest_s` , MAX(  `senyawa_s` ) AS  `highest_s`, MIN(
14. `senyawa_f` ) AS  `lowest_f` , MAX(  `senyawa_f` ) AS
15. `highest_f`, MIN(  `senyawa_cl` ) AS  `lowest_cl` , MAX(
16. `senyawa_cl` ) AS  `highest_cl`, MIN(  `senyawa_br` ) AS
17. `lowest_br` , MAX(  `senyawa_br` ) AS  `highest_br`, MIN(
18. `senyawa_i` ) AS  `lowest_i` , MAX(  `senyawa_i` ) AS
19. `highest_i`, MIN(  `senyawa_oh` ) AS  `lowest_oh` , MAX(
20. `senyawa_oh` ) AS  `highest_oh` FROM  tbl_senyawa" ) or die
21. (mysql_error());
22.                   while ($row= mysql_fetch_array ($result) ){
23.
24.                         ?>
25.
26.                         <tr>
27.
28.                               <td><?php echo "Jarak"; ?></td>
29.                               <td><?php echo $row['highest_b']
30. - $row['lowest_b'];$jarak_b = $row['highest_b'] -
31. $row['lowest_b'];?></td>
32.                               <td><?php echo $row['highest_c']
33. - $row['lowest_c'];$jarak_c = $row['highest_c'] -
34. $row['lowest_c']; ?></td>
```

```
35.                                 <td><?php echo $row['highest_n']
36.   - $row['lowest_n'];$jarak_n = $row['highest_n'] -
37.   $row['lowest_n']; ?></td>
38.                                 <td><?php echo $row['highest_o']
39.   - $row['lowest_o'];$jarak_o = $row['highest_o'] -
40.   $row['lowest_o']; ?></td>
41.                                 <td><?php echo $row['highest_p']
42.   - $row['lowest_p'];$jarak_p = $row['highest_p'] -
43.   $row['lowest_p']; ?></td>
44.                                 <td><?php echo $row['highest_s']
45.   - $row['lowest_s'];$jarak_s = $row['highest_s'] -
46.   $row['lowest_s']; ?></td>
47.                                 <td><?php echo $row['highest_f']
48.   - $row['lowest_f'];$jarak_f = $row['highest_f'] -
49.   $row['lowest_f']; ?></td>
50.                                 <td><?php echo
51.   $row['highest_cl'] - $row['lowest_cl'];$jarak_cl =
52.   $row['highest_cl'] - $row['lowest_cl']; ?></td>
53.                                 <td><?php echo
54.   $row['highest_br'] - $row['lowest_br'];$jarak_br =
55.   $row['highest_br'] - $row['lowest_br']; ?></td>
56.                                 <td><?php echo $row['highest_i']
57.   - $row['lowest_i'];$jarak_i = $row['highest_i'] -
58.   $row['lowest_i']; ?></td>
59.                                 <td><?php echo
60.   $row['highest_oh'] - $row['lowest_oh'];$jarak_oh =
61.   $row['highest_oh'] - $row['lowest_oh']; ?></td>
62.                                 <td>-</td>
63.
64.
```

Source Code 5.2 Kode Program Nilai Maksimum, Minimum dan Jarak

Penjelasan:

Baris 1-3 : Untuk proses inisialisasi

Baris 5-21 : Untuk mencari nilai terkecil dan terbesar dari *dataset*

Baris 28-64 : Untuk mencari nilai jarak dan memasukkan nilainya pada tabel

5.3.2.2 Pengurutan Data Berdasar Kolom Jarak Terbesar

Proses selanjutnya yaitu mengurutkan data dari yang terkecil berdasarkan kolom dengan jarak terbesar. Proses pengurutan data dapat dilihat pada Source Code 5.3.

```
1.   $jarak_last = 0;
2.
3.                      if($jarak_b > $jarak_last){
4.                            $jarak_last = $jarak_b;
5.                            $jarak_alpha = "senyawa_b";
6.                      }
7.                      if($jarak_c > $jarak_last){
8.                            $jarak_last = $jarak_c;
9.                            $jarak_alpha = "senyawa_c";
10.                     }
11.                     if($jarak_n > $jarak_last){
12.                           $jarak_last = $jarak_n;
```

```
13.                         $jarak_alpha = "senyawa_n";
14.                   }
15.                   if($jarak_o > $jarak_last){
16.                         $jarak_last = $jarak_o;
17.                         $jarak_alpha = "senyawa_o";
18.                   }
19.                   if($jarak_p > $jarak_last){
20.                         $jarak_last = $jarak_p;
21.                         $jarak_alpha = "senyawa_p";
22.                   }
23.                   if($jarak_s > $jarak_last){
24.                         $jarak_last = $jarak_s;
25.                         $jarak_alpha = "senyawa_s";
26.                   }
27.                   if($jarak_f > $jarak_last){
28.                         $jarak_last = $jarak_f;
29.                         $jarak_alpha = "senyawa_f";
30.                   }
31.                   if($jarak_cl > $jarak_last){
32.                         $jarak_last = $jarak_cl;
33.                         $jarak_alpha = "senyawa_cl";
34.                   }
35.                   if($jarak_br > $jarak_last){
36.                         $jarak_last = $jarak_br;
37.                         $jarak_alpha = "senyawa_br";
38.                   }
39.                   if($jarak_i > $jarak_last){
40.                         $jarak_last = $jarak_i;
41.                         $jarak_alpha = "senyawa_i";
42.                   }
43.                   if($jarak_oh > $jarak_last){
44.                         $jarak_last = $jarak_oh;
45.                         $jarak_alpha = "senyawa_oh";
46.                   }
47.
48.                   echo "<b>Jarak Terbesar: </b>".$jarak_last."
49. (".$jarak_alpha.") <br><br> ";
50.                   $_SESSION['tertinggi'] = $jarak_alpha;
```

Source Code 5.3 Pengurutan Data Berdasar Kolom Jarak Terbesar

Penjelasan:

Baris 1 : Inisialisasi variabel yang digunakan untuk melakukan pengecekan nilai jarak terbesar

Baris 3-46 : Penelusuran tiap data untuk mencari nilai jarak terbesar

Baris 48-49 : Untuk menampilkan nilai jarak terbesar

5.3.2.3 Membagi Data Sejumlah 'K'

Proses selanjutnya data yang sudah diurutkan dibagi sejumlah 'K' klaster sama banyak. Proses pembagian data sejumlah 'K' dapat dilihat pada Source Code 5.4.

```
1.  <?php
2.                   $data_no =1;
3.                   $data_dibagi = 1;
4.
```

```
5.                      $result=mysql_query("SELECT count(*) as
6.    total from tbl_senyawa");
7.                      $total_data=mysql_fetch_assoc($result);
8.
9.                      $ctr_pembagian = $total_data['total'] / 7;
10.
11.                     $ctr_satusampaitujuh = 1;
12.
13.               ?>
14.               <?php
15.               $query = "SELECT * from tbl_senyawa2";
16.               $result = mysql_query($query);
17.
18.               if(mysql_num_rows($result) == 0)
19.               {
20.                     $result= mysql_query("select * from
21.    tbl_senyawa ORDER BY CAST(".$jarak_alpha." AS
22.    DECIMAL(10,10))") or die (mysql_error());
23.                     while ($row= mysql_fetch_array ($result) ){
24.
25.                           if($data_dibagi /
26.    round($ctr_pembagian) == 1){
27.                                 $ctr_satusampaitujuh++;
28.                                 $data_dibagi = 1;
29.                           }
30.                           if($ctr_satusampaitujuh == 2){
31.                                 $ctr_satusampaitujuh--;
32.                           }
33.                           $query=mysql_query("INSERT INTO
34.    tbl_senyawa2(id_senyawa,id_senyawa_datake,senyawa_smiles,senya
35.    wa_b,senyawa_c,senyawa_n,senyawa_o,senyawa_p,senyawa_s,senyawa
36.    _f,senyawa_cl,senyawa_br,senyawa_i,senyawa_oh,senyawa_penyakit
37.    )VALUES('".$data_no."','".$row['id_senyawa']."','".$row['senya
38.    wa_smiles']."','".$row['senyawa_b']."','".$row['senyawa_c']."'
39.    ,'".$row['senyawa_n']."','".$row['senyawa_o']."','".$row['seny
40.    awa_p']."','".$row['senyawa_s']."','".$row['senyawa_f']."','".
41.    $row['senyawa_cl']."','".$row['senyawa_br']."','".$row['senyaw
42.    a_i']."','".$row['senyawa_oh']."','".$ctr_satusampaitujuh."')"
43.    ) or die(mysql_error());
44.
45.                           $data_dibagi++;
46.                           $data_no++;
47.                     }
48.               }
```

Source Code 5.4 Membagi Data Sejumlah 'K'

Penjelasan:

Baris 2-3 : Inisialisasi variabel

Baris 5-11 : Proses menghitung jumlah data dan dibagi sebanyak jumlah K

Baris 20-46 : Proses inisialisasi kelas 1-2 dari data yang sudah dibagi sejumlah K dan memasukkan ke dalam tabel

5.3.2.4 Menghitung Pusat Klaster Awal Heuristic O(n logn)

Proses selanjutnya yaitu menghitung pusat klaster awal yaitu dengan mencari rata-rata dari masing-masing 'K' bagian, kemudian di set sebagai pusat klaster awal. Proses perhitungan pusat klaster dapat dilihat pada Source Code 5.5.

```
1.  $result= mysql_query("SELECT senyawa_penyakit, AVG(senyawa_b)
2.  AS average_b, AVG(senyawa_c) AS average_c, AVG(senyawa_n) AS
3.  average_n, AVG(senyawa_o) AS average_o, AVG(senyawa_p) AS
4.  average_p, AVG(senyawa_s) AS average_s, AVG(senyawa_f) AS
5.  average_f, AVG(senyawa_cl) AS average_cl, AVG(senyawa_br) AS
6.  average_br, AVG(senyawa_i) AS average_i, AVG(senyawa_oh) AS
7.  average_oh FROM tbl_senyawa2 WHERE senyawa_penyakit='1'");
8.
9.                          $row = mysql_fetch_assoc($result);
10.                         echo "<tr>";
11.                         echo "<td>C".$ctr_centroid."</td>";
12.                         echo "<td>".$row['average_b']."</td>";
13.                         echo "<td>".$row['average_c']."</td>";
14.                         echo "<td>".$row['average_n']."</td>";
15.                         echo "<td>".$row['average_o']."</td>";
16.                         echo "<td>".$row['average_p']."</td>";
17.                         echo "<td>".$row['average_s']."</td>";
18.                         echo "<td>".$row['average_f']."</td>";
19.                         echo
20. "<td>".$row['average_cl']."</td>";
21.                         echo
22. "<td>".$row['average_br']."</td>";
23.                         echo "<td>".$row['average_i']."</td>";
24.                         echo
25. "<td>".$row['average_oh']."</td>";
26.                         echo
27. "<td>".$row['senyawa_penyakit']."</td>";
28.                         echo "</tr>";
29.
30.                         $query=mysql_query("INSERT INTO
31. tbl_randompick2(id_senyawa,senyawa_b,senyawa_c,senyawa_n,senya
32. wa_o,senyawa_p,senyawa_s,senyawa_f,senyawa_cl,senyawa_br,senya
33. wa_i,senyawa_oh,senyawa_penyakit)VALUES('".$ctr_centroid."','"
34. .$row['average_b']."','".$row['average_c']."','".$row['average
35. _n']."','".$row['average_o']."','".$row['average_p']."','".$ro
36. w['average_s']."','".$row['average_f']."','".$row['average_cl'
37. ]."','".$row['average_br']."','".$row['average_i']."','".$row[
38. 'average_oh']."','".$ctr_centroid."')") or die(mysql_error());
```

Source Code 5.5 Menghitung Pusat Klaster Awal *Heuristic O(n logn)*

Penjelasan:

Baris 1-7 : Mengambil data yang akan dihitung rata-ratanya

Baris 9-28 : Mencari nilai rata-rata tiap elemen

Baris 30-38 : Memasukkan nilai hasil perhitungan rata-rata ke dalam tabel

5.3.3 Perhitungan K-Means

5.3.3.1 Menghitung Jarak Pusat Klaster dengan *Dataset*

Proses selanjutnya yaitu menghitung jarak menggunakan *euclidean distance*. Proses ini akan digunakan pada metode K-Means maupun *improved* K-Means. Proses perhitungan jarak pusat klaster dengan *dataset* dapat dilihat pada Source Code 5.6.

```
1.  if($ctr_tabel_x == $_SESSION['jumlah_tabel_a']){
2.
3.                                  $ctr_tabel_x = 0;
4.
5.                          }
6.
7.                          if($ctr_tabel_y ==
8.  $_SESSION['jumlah_tabel_b']){
9.
10.                                 $ctr_tabel_y = 0;
11.
12.                         }
13.
14.                         for ($y = 0; $y <= 10; $y++) {
15.
16.                                 $hasil = $hasil +
17. pow($tabel_a[$ctr_tabel_x][$y] -
18. $tabel_b[$ctr_tabel_y][$y],2);
19.
20.                                 }
21.
22.                         $hasilakhir = sqrt($hasil);
23.
24.                         echo $hasilakhir;
25.
26.
27.
28.                         $Simpan_arr[$ctr_tabel_x][] =
29. $hasilakhir;
30.
31.
32.
33.                         $hasil =0;
34.
35.                         $hasilakhir =0;
36.
37.                         $ctr_tabel_y++;
```

Source Code 5.6 Menghitung Jarak Pusat Klaster dengan *Dataset*

Penjelasan:

Baris 1-24 : Proses mencari jarak tiap data dengan pusat klaster menggunakan rumus *euclidean distance*

5.3.3.5 Menghitung Pusat Klaster Baru

Proses selanjutnya yaitu mencari pusat klaster baru dengan menghitung nilai rata-rata dari data yang masuk ke masing-masing klaster. Pusat klaster baru

digunakan untuk menghitung jarak pada iterasi selanjutnya. Proses perhitungan pusat klaster baru dapat dilihat pada Source Code 5.7.

```
1.  $jumlah_cluster_b = $jumlah_cluster_b / $ctr_jumlah_cluster;
2.
3.                                      $jumlah_cluster_c =
4.  $jumlah_cluster_c / $ctr_jumlah_cluster;
5.
6.                                      $jumlah_cluster_n =
7.  $jumlah_cluster_n / $ctr_jumlah_cluster;
8.
9.                                      $jumlah_cluster_o =
10. $jumlah_cluster_o / $ctr_jumlah_cluster;
11.
12.                                     $jumlah_cluster_p =
13. $jumlah_cluster_p / $ctr_jumlah_cluster;
14.
15.                                     $jumlah_cluster_s =
16. $jumlah_cluster_s / $ctr_jumlah_cluster;
17.
18.                                     $jumlah_cluster_f =
19. $jumlah_cluster_f / $ctr_jumlah_cluster;
20.
21.                                     $jumlah_cluster_cl =
22. $jumlah_cluster_cl / $ctr_jumlah_cluster;
23.
24.                                     $jumlah_cluster_br =
25. $jumlah_cluster_br / $ctr_jumlah_cluster;
26.
27.                                     $jumlah_cluster_i =
28. $jumlah_cluster_i / $ctr_jumlah_cluster;
29.
30.                                     $jumlah_cluster_oh =
31. $jumlah_cluster_oh / $ctr_jumlah_cluster;
32.
33.
34.       $queryupdate=mysql_query("UPDATE tbl_randompick SET
35. senyawa_b='".$jumlah_cluster_b."',senyawa_c='".$jumlah_cluster
36. _c."',senyawa_n='".$jumlah_cluster_n."',senyawa_o='".$jumlah_c
37. luster_o."',senyawa_p='".$jumlah_cluster_p."',senyawa_s='".$ju
38. mlah_cluster_s."',senyawa_f='".$jumlah_cluster_f."',senyawa_cl
39. ='".$jumlah_cluster_cl."',senyawa_br='".$jumlah_cluster_br."',
40. senyawa_i='".$jumlah_cluster_i."',senyawa_oh='".$jumlah_cluste
41. r_oh."' WHERE id_senyawa='".$ctr_id_tabel."'") or
42. die(mysql_error());
43.                                     $ctr_id_tabel++;
```

Source Code 5.7 Menghitung Pusat Klaster Baru

Penjelasan:

Baris 1-31 : Mencari nilai rata-rata tiap klaster

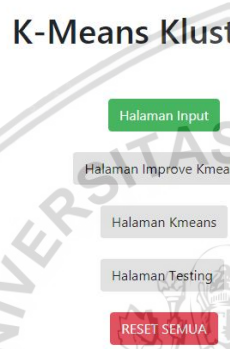
Baris 34-42 : Mengganti pusat klaster lama dengan pusat klaster baru pada tabel

5.4 Implementasi Antarmuka

Implementasi antarmuka berisi gambaran sistem yang telah dibuat beserta tahapan-tahapan implementasi algoritme K-Means dengan inisial pusat klaster menggunakan metode *heuristic o(n logn)*. Antarmuka sistem merujuk pada perancangan yang telah dibuat.

5.4.1 Halaman Awal

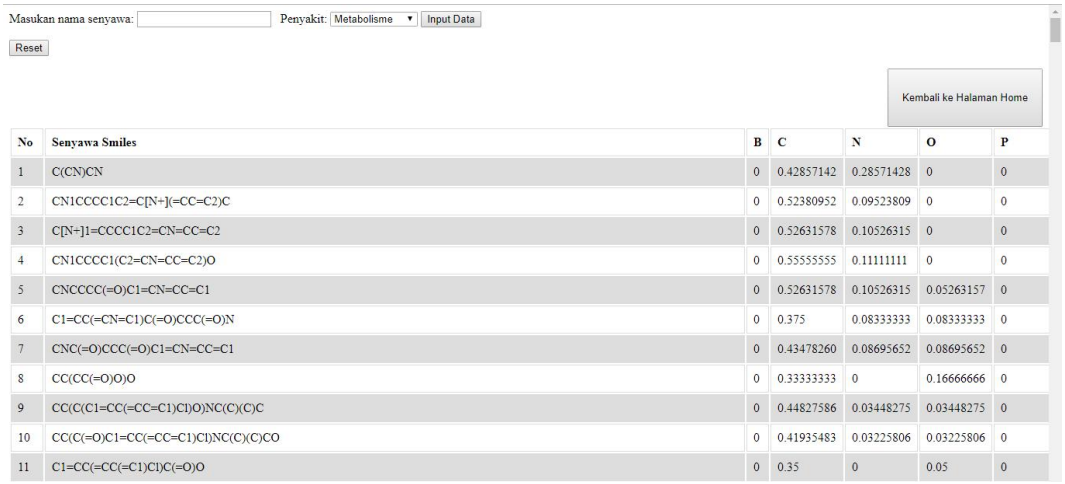
Antarmuka halaman awal menampilkan judul sistem, menu bar yang terdiri dari *button input* data, *button improved* K-Means, *button* K-Means dan *button Testing.* Hasil implementasi antarmuka halaman awal ditunjukkan pada Gambar 5.1.



Gambar 5.1 Implementasi Halaman Awal

5.4.2 Halaman *Input* Data

Antarmuka halaman *input* data menampilkan kolom untuk memasukkan senyawa, kemudian memilih jenis data yang sesuai dengan jenis penyakit, lalu tekan *button input* data. Data yang sudah dimasukkan akan tersimpan di *database* dan akan ditampilkan pada tabel. Kemudian kembali ke *button* Home untuk melanjutkan proses pelatihan. Hasil implementasi antarmuka *input* data dapat dilihat pada Gambar 5.2

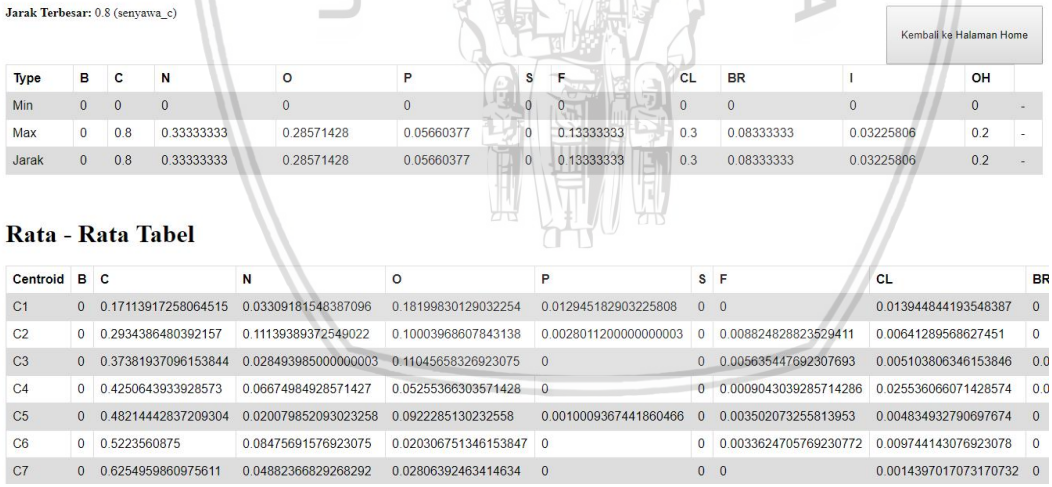


No	Senyawa Smiles	B	C	N	O	P
1	C(CN)CN	0	0.42857142	0.28571428	0	0
2	CN1CCCC1C2=C[N+](=CC=C2)C	0	0.52380952	0.09523809	0	0
3	C[N+]1=CCCC1C2=CN=CC=C2	0	0.52631578	0.10526315	0	0
4	CN1CCCC1(C2=CN=CC=C2)O	0	0.55555555	0.11111111	0	0
5	CNCCCC(=O)C1=CN=CC=C1	0	0.52631578	0.10526315	0.05263157	0
6	C1=CC(=CN=C1)C(=O)CCC(=O)N	0	0.375	0.08333333	0.08333333	0
7	CNC(=O)CCC(=O)C1=CN=CC=C1	0	0.43478260	0.08695652	0.08695652	0
8	CC(CC(=O)O)O	0	0.33333333	0	0.16666666	0
9	CC(C(C1=CC(=CC=C1)Cl)O)NC(C)(C)C	0	0.44827586	0.03448275	0.03448275	0
10	CC(C(=O)C1=CC(=CC=C1)Cl)NC(C)(C)CO	0	0.41935483	0.03225806	0.03225806	0
11	C1=CC(=CC(=C1)Cl)C(=O)O	0	0.35	0	0.05	0

Gambar 5.2 Implementasi Halaman *Input* Data

5.4.3 Halaman *Training Improved* K-Means

Pada halaman *Improved* K-Means sistem akan langsung memroses data yang telah dimasukkan pada halaman *input* data. Hasil keluaran berupa jarak, dan hasil akhir iterasi berupa pusat klaster awal yang akan digunakan pada proses pengujian. Kemudian tekan *button* Home untuk kembali ke halaman awal dan melanjutkan proses. Hasil implementasi antarmuka halaman *improved* K-Means dapat dilihat pada Gambar 5.3.



Type	B	C	N	O	P	S	F	CL	BR	I	OH	
Min	0	0	0	0	0	0	0	0	0	0	0	-
Max	0	0.8	0.33333333	0.28571428	0.05660377	0	0.13333333	0.3	0.08333333	0.03225806	0.2	-
Jarak	0	0.8	0.33333333	0.28571428	0.05660377	0	0.13333333	0.3	0.08333333	0.03225806	0.2	-

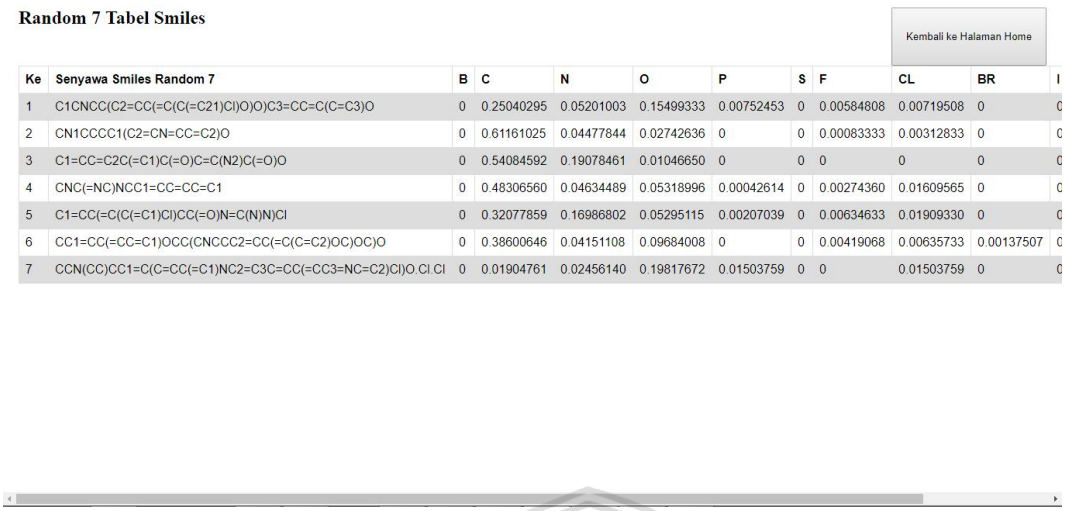
Rata - Rata Tabel

Centroid	B	C	N	O	P	S	F	CL	BR
C1	0	0.17113917258064515	0.03309181548387096	0.18199830129032254	0.012945182903225808	0	0	0.013944844193548387	0
C2	0	0.2934386480392157	0.11139389372549022	0.10003968607843138	0.002801120000000003	0	0.008824828823529411	0.00641289568627451	0
C3	0	0.37381937096153844	0.028493985000000003	0.11045658326923075	0	0	0.005635447692307693	0.005103806346153846	0.0
C4	0	0.4250643933928573	0.06674984928571427	0.05255366303571428	0	0	0.009043039285714286	0.025536066071428574	0.0
C5	0	0.48214442837209304	0.020079852093023258	0.0922285130232558	0.0010009367441860466	0	0.003502073255813953	0.004834932790697674	0
C6	0	0.5223560875	0.08475691576923075	0.020306751346153847	0	0	0.003362470576923077	0.009744143076923078	0
C7	0	0.6254959860975611	0.04882366829268292	0.02806392463414634	0	0	0	0.0014397017073170732	0

Gambar 5.3 Implementasi Halaman *Improved* K-Means

5.4.4 Halaman *Training* K-Means

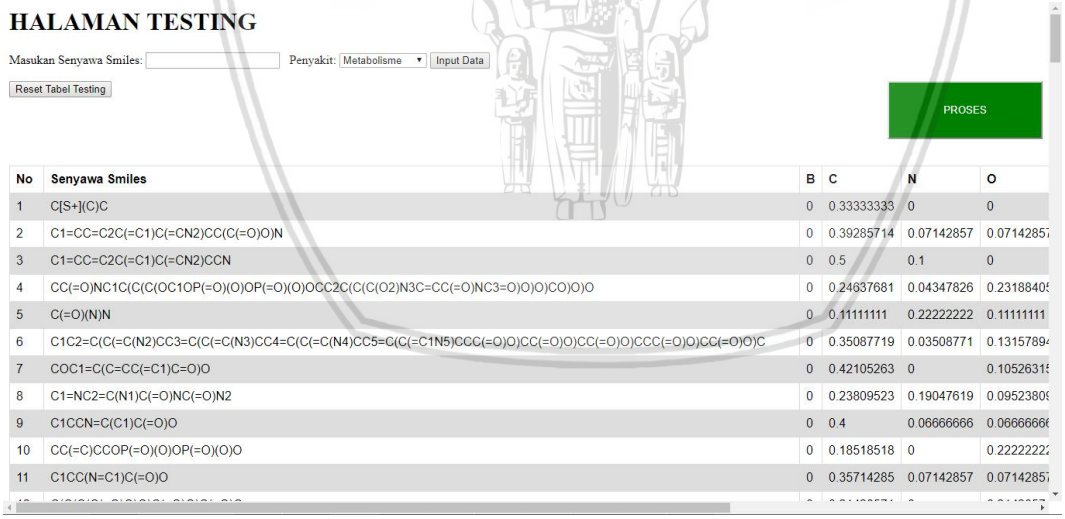
Pada halaman K-Means sistem akan memproses data latih dan menampilkan hasil berupa pusat klaster awal yang telah didapatkan dari proses iterasi yang telah konvergen. Hasil implementasi antarmuka halaman K-Means dapat dilihat pada Gambar 5.4.



Gambar 5.4 Implementasi Halaman K-Means

5.4.5 Halaman Pengujian

Pada halaman pengujian, sistem menampilkan halaman *input* data uji. Hasil implementasi *input* data uji ditunjukkan pada Gambar 5.5. Sedangkan *button* proses dapat langsung memroses data uji dengan pusat klaster yang telah didapatkan dari proses pelatihan. Halaman proses pengujian akan menampilkan akurasi dan banyaknya iterasi dari proses pengujian. Hasil implementasi halaman proses pengujian ditunjukkan pada Gambar 5.6



Gambar 5.5 Implementasi Halaman *Input* Data Uji

Tampilkan Proses

Kembali ke Halaman Home

- Akurasi Improved K-Means: 40%

- Akurasi K-Means: 30%

Gambar 5.6 Implementasi Halaman Proses Pengujian

Tampilkan Proses

Kembali ke Halaman Home

Jarak ke-Centroid Dengan Rata-Rata 7

Data ke-i	1	2	Terdekat	Cluster
Data x1	0.087154664776588	0.28831776122809	0.087154664776588	1
Data x2	0.12587346208439	0.2218630774158	0.12587346208439	1
Data x3	0.24248915581335	0.083435937884349	0.083435937884349	2
Data x4	0.33502406621532	0.44051127700897	0.33502406621532	1
Data x5	0.30818536273452	0.11230012002903	0.11230012002903	2
Data x6	0.12279472048286	0.2050198494826	0.12279472048286	1
Data x7	0.1065710194783	0.13795096526866	0.1065710194783	1
Data x8	0.15043391353546	0.16609912743327	0.15043391353546	1
Data x9	0.15822812157343	0.15674505909201	0.15674505909201	2
Data x10	0.09504455380151	0.15785615218046	0.09504455380151	1

Gambar 5.7 Implementasi Halaman Tampilkan Proses Pengujian

BAB 6 PEMBAHASAN

6.1 Pengujian dan Analisis

Proses pengujian metode *K-Means* dengan inisial pusat klaster menggunakan metode *heuristic o(n logn)* dilakukan terhadap jumlah data terhadap nilai akurasi dan jumlah iterasi. Terdapat beberapa skenario pengujian yang dilakukan pada variasi jumlah data. Pengujian dilakukan untuk membandingkan hasil antara K-Means dan *Improved* K-Means.

6.1.1 Pengujian Validitas Program

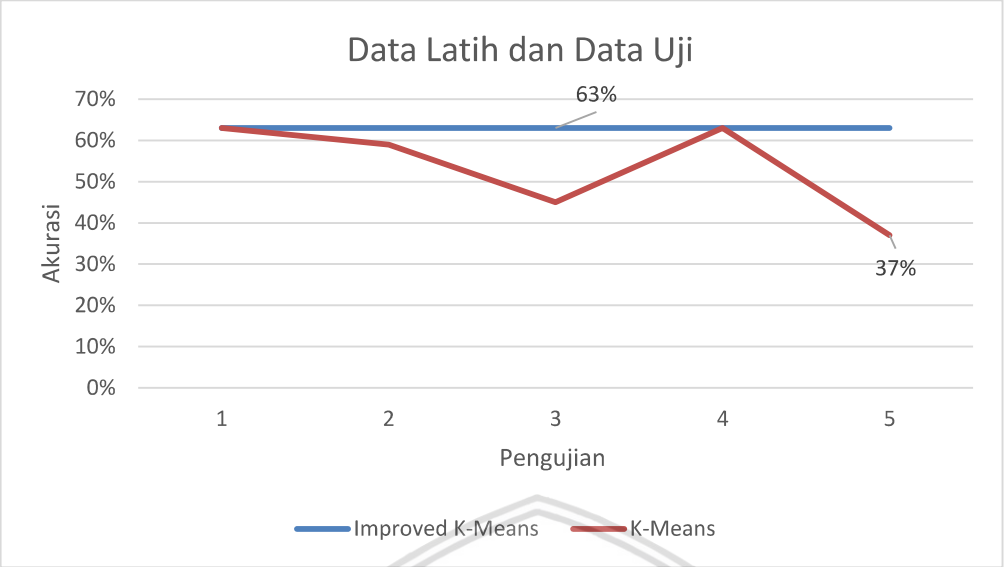
Pengujian validitas untuk mengetahui kesesuaian program dengan perhitungan manual dilakukan menggunakan data yang sama dengan manualisasi yaitu sebanyak 20 data uji dan 10 data latih. Data yang digunakan berjumlah dua kelas yaitu kelas kanker dan metabolisme. Hasil akurasi yang didapat oleh program sama dengan hasil perhitungan manual yaitu sebesar 40%.

6.1.2 Pengujian Data Latih dan Data Uji

Pengujian data latih dan data uji dilakukan dengan menggunakan seluruh data yang ada yaitu sebanyak 512 data latih dan 128 data uji. Data latih terdiri dari 350 data metabolisme dan 162 data kanker, sedangkan data uji terdiri dari 87 data metabolisme dan 41 data kanker. Hasil pengujian yang dilakukan sebanyak lima kali mendapatkan nilai akurasi rata-rata *improved* K-Means sebesar 63%, sedangkan rata-rata K-Means sebesar 53,4%. Hasil pengujian dapat dilihat pada Tabel 6.1.

Tabel 6.1 Hasil Pengujian Data Latih dan Data Uji

Metode	1	2	3	4	5	Rata-rata
Improved K-Means	63%	63%	63%	63%	63%	63%
K-Means	63%	59%	45%	63%	37%	53,4%

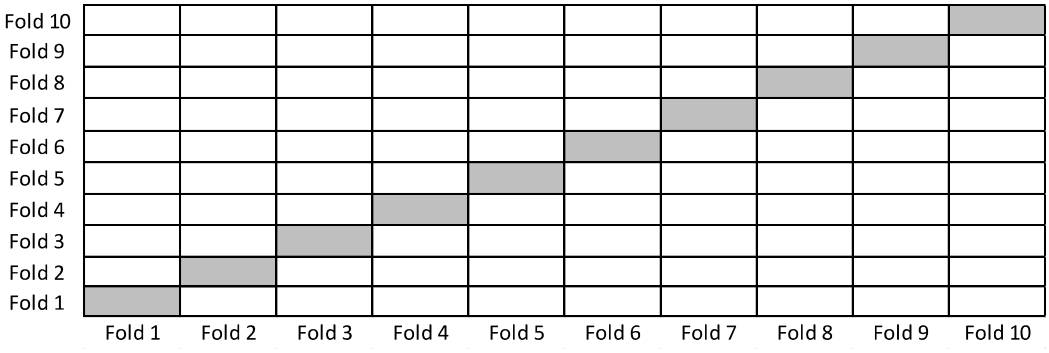


Gambar 6.1 Grafik Hasil Pengujian Data Latih dan Data Uji

Berdasarkan grafik pengujian pada Gambar 6.1 yang didapat dari lima kali pengujian menggunakan data latih dan data uji yang sama, didapat kesimpulan bahwa metode *improved* K-Means mempunyai hasil yang konstan, sedangkan K-Means mempunyai hasil yang berbeda. Hal ini dikarenakan inisialisasi yang dilakukan pada K-Means bersifat *random*, sehingga dapat bergantung pada inisial yang dilakukan. Berbeda dengan *improved* K-Means yang inisialisasi pusat klaster awalnya dilakukan dengan menggunakan metode *heuristic o(n logn)*, hasil yang didapat tidak berubah, sehingga akurasi maksimum bisa langsung diketahui.

6.1.3 Pengujian *K-Fold Cross Validation*

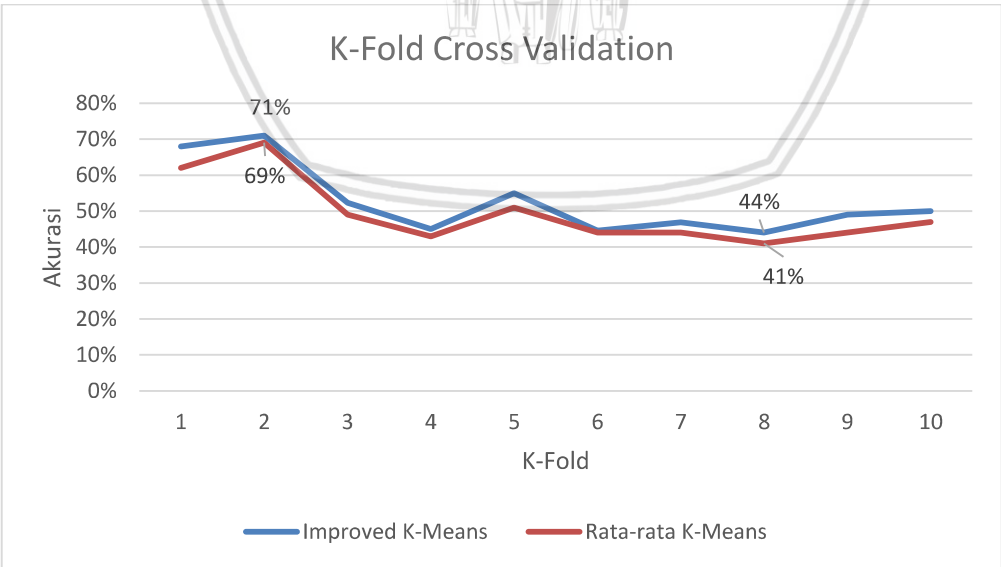
K-Fold Cross Validation merupakan salah satu metode yang bertujuan untuk mengukur tingkat keakuratan sistem terhadap data yang diuji. Pada penelitian ini *dataset* dibagi menjadi 10 bagian dari keseluruhan data. Setiap bagian akan menjadi data uji dan dilakukan pengujian masing-masing, sedangkan bagian yang tidak menjadi data uji akan menjadi data latih. Pengujian ini bertujuan mengetahui keakuratan sistem terhadap data yang bersifat *random.* Data yang digunakan terdiri dari 574 data latih yang meliputi 182 data kanker dan 392 data metabolisme, serta 65 data uji yang meliputi 21 data kanker dan 44 data metabolisme. Pengujian dilakukan sebanyak tiga kali dengan masing-masing pengujian terdapat 10 Fold data. Gambar 6.2 menunjukkan skenario pembagian data set setiap *Fold*. Sedangkan Tabel 6.2 dan Gambar 6.3 merupakan hasil dari pengujian *K-Fold Cross Validation.*



Gambar 6.2 Pembagian *Dataset*

Tabel 6.2 Hasil Pengujian *K-Fold Cross Validation*

Metode	K-Fold										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
Improved K-Means	68%	71%	52,3 %	45%	55%	44,6 %	46,9 %	44%	49 %	50%	52,58%
K-Means 1	50%	69%	51%	44%	53%	43%	45%	37%	49 %	47%	49%
K-Means 2	68%	70%	45%	40%	50%	44,6 %	39%	42%	39, 8%	45%	48%
K-Means 3	68%	68%	50,7 %	45%	50%	44,6 %	46,9 %	44%	42, 8%	50%	51%
Rata-rata	62%	69%	49%	43%	51%	44%	44%	41%	44 %	47%	49%



Gambar 6.3 Grafik Hasil Pengujian *K-Fold Cross Validation*

Berdasarkan grafik pengujian *K-Fold Cross Validation* yang ditunjukkan pada Gambar 6.3 dapat disimpulkan bahwa *Improved K-Means* mempunyai hasil akurasi

yang lebih baik yaitu dengan rata-rata sebesar 52,58% dibandingkan K-Means dengan rata-rata sebesar 49%. Hasil yang didapatkan menunjukkan bahwa metode K-Means konvensional mendapatkan hasil yang berbeda/berubah selama tiga kali pengujian. Sedangkan *Improved* K-Means memperoleh hasil yang optimal dan tidak berubah. Dengan pengujian sebanyak 10 *Fold*, masing-masing hasil akurasi menunjukkan bahwa *Improved K-Means* mempunyai akurasi sama atau lebih baik dari pada K-Means Konvensional. Akurasi terbesar *Improved K-Means* yaitu pada *Fold* ke-2 sebesar 71% dan akurasi terkecil pada *Fold* ke-8 sebesar 44%. Sedangkan pada K-Means akurasi terbesar pada *Fold* ke-2 pengujian ke-2 yaitu 70% dan akurasi terkecil pada *Fold* ke-8 pengujian ke-1 sebesar 37%.

BAB 7 PENUTUP

Berdasarkan penelitian yang dilakukan, maka diperoleh kesimpulan dan saran yang akan dijelaskan pada bab ini. Kesimpulan dan saran dapat digunakan sebagai masukan untuk penelitian selanjutnya.

7.1 Kesimpulan

Kesimpulan yang diperoleh berdasarkan penelitian mengenai pengelompokan fungsi aktif senyawa aktif SMILES menggunakan metode K-Means dengan inisialisasi pusat klaster menggunakan metode *Heuristic O(N LogN)* adalah:

1. Penerapan metode *heuristic o(n logn)* untuk inisialisasi pusat klaster K-Means yang pertama didapatkan dengan mencari nilai maksimum dan minimum setiap atribut *dataset*. Kedua, data diurutkan berdasarkan kolom yang mempunyai nilai rentang terbesar. Langkah ketiga, data dibagi menjadi 'k' bagian sama banyak, kemudian dihitung nilai rata-rata tiap bagian untuk ditetapkan sebagai nilai pusat klaster awal.
2. Penelitian dengan menerapkan inisial pusat klaster K-Means menggunakan metode *heuristic o(n logn)* pada senyawa aktif kode SMILES mampu meningkatkan akurasi lebih baik dibandingkan K-Means konvensional. Pengujian dilakukan menggunakan dua kelas *dataset* yaitu kelas kanker dan metabolisme. Hasil yang didapatkan meliputi:
    a. Pengujian validasi program yang bertujuan untuk mengetahui kesesuaian program dengan perhitungan manual sudah sama. Hasil akurasi yang diperoleh dari pengujian validasi program yaitu sebesar 40%.
    b. Pengujian data latih dan data uji dengan menggunakan seluruh data latih dan data uji yaitu sebanyak 512 data dan 128 data memperoleh hasil akurasi *Improved* K-Means sebesar 63% dan K-Means sebesar 53,4%.
    c. Pengujian *K-Fold Cross Validation* pada *improved K-Means* memperoleh rata-rata akurasi sebesar 52,58% dari 10 kali percobaan. Sedangkan rata-rata akurasi untuk K-Means sebesar 49%.

7.2 Saran

Berdasarkan penelitian yang telah dilakukan, saran yang dapat diberikan untuk penelitian selanjutnya adalah:

Data pada notasi SMILES bersifat acak dan kurang menunjukkan karakteristik pada tiap kelasnya. Hal ini ditunjukkan dengan adanya notasi yang tidak terdapat dalam pemilihan fitur. Untuk meningkatkan akurasi, dapat dilakukan pemilihan data dan kelas yang berpengaruh pada fitur yang digunakan. Disarankan juga untuk melakukan penelitian dengan metode lain.

DAFTAR PUSTAKA

Manning, C., P. Raghavan, dan H. Schutze. 2009. *Introduction to Information Retrieval.* England: Cambridge University Press.

Salni, H. Marisa., dan R. W. Mukti. 2011. Isolasi Senyawa Antibakteri Dari Daun Jengkol (Pithecolobium Iobatum Benth) dan Penentuan Nilai KHM-nya. *Jurnal Penelitian Sains,* Volume 14, pp. 38-41.

Hermawati, F. A., 2013. *Data Mining.* Yogyakarta: Penerbit Andi.

Junaedi, H. 2011. Penggambaran Rantai Karbon Dengan Menggunakan Simplified Molecular Input Line System (SMILES). *Sekolah Tinggi Teknik Surabaya.*

Mukhriani. 2016. Ekstraksi, Pemisahan Senyawa, dan Identifikasi Senyawa Aktif. Program Studi Farmasi Fakultas Ilmu Kesehatan UIN Alauddin Makassar.

Nazeer, K. A. A., S. M. Kumar, dan M. P. Sebastian. 2011. Enhancing the K-means Clustering Algorithm by Using a O(n logn) Heuristic Method for Finding Better Initial Centroids. *International Conference on Emerging Applications of Information Technology- EAIT 2011.*

Rizki, M. I., dan E. M. Hariandja. 2015. Aktivitas Farmakologis, Senyawa Aktif, dan Mekanisme Kerja Daun Salam (Syzygium Polyanthum). *Prosiding Seminar Nasional & Workshop "Perkembangan Terkini Sains Farmasi & Klinik",* Volume 5, pp. 239-244.

Prasetyo, E. 2012. *Klasifikasi Metode-Metode Pilihan.* Yogyakarta: Penerbit Andi.

Astuti, R., D. E. Ratnawati., dan B. D. Setiawan. 2015. Implementasi Algoritme K-Means Clustering dengan Inisialisasi Centroid Menggunakan Metode Heuristic O(N LogN). *Repositori Jurnal Mahasiswa PTIIK UB,* 6(17).

Santosa, B. 2007. *Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis, Teori dan Aplikasi.* Yogyakarta: Graha Ilmu.

Alfina, T., B. Santosa, dan A. R. Barakbah. 2012. Analisa Perbandingan Metode Hierarchical Clustering,K-Means dan Gabungan Keduanya Dalam Membentuk Cluster Data (Studi Kasus : Problem Kerja Praktek Jurusan Teknik Industri ITS). *Jurnal Teknik POMITS,* Volume 1, pp. 1-5.

Weininger, D. 1988. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.,* pp. 31-36.